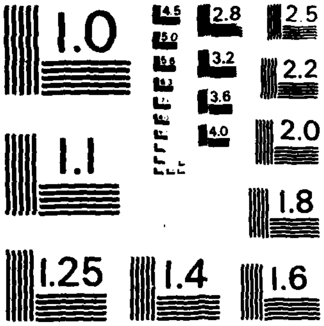
AD-A139 339   I&W (INDICATIONS AND WARNING) DATA BASE MANAGEMENT   1/4
SYSTEMS ANALYSIS(U) MEASUREMENT CONCEPT CORP ROME NY
JUN 83 RADC-TR-83-136 F30602-80-C-0286

UNCLASSIFIED   F/G 5/2   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(12)

RADC-TR-83-136
Final Technical Report
June 1983

I&W DATA BASE MANAGEMENT SYSTEMS ANALYSIS

Measurement Concept Corporation

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-83-136 has been reviewed and is approved for publication.

APPROVED: Patrick K McCabe

PATRICK K. MCCABE
Project Engineer

APPROVED: Thadeus Domurat

THADEUS J. DOMURAT
Acting Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRDA) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document requires that it be returned.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER RADC-TR-83-136	2. GOVT ACCESSION NO. AD-A139339	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) I&W DATA BASE MANAGEMENT SYSTEMS ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED Final Technical Report 15 Sep 80 - 15 Dec 82
		6. PERFORMING ORG. REPORT NUMBER N/A
7. AUTHOR(s) Measurement Concept Corporation		8. CONTRACT OR GRANT NUMBER(s) F30602-80-C-0286
9. PERFORMING ORGANIZATION NAME AND ADDRESS Measurement Concept Corporation 1721 Black River Blvd Rome NY 13440		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 64750F 19550708
11. CONTROLLING OFFICE NAME AND ADDRESS Rome Air Development Center (IRDA) Griffiss AFB NY 13441		12. REPORT DATE June 1983
		13. NUMBER OF PAGES 348
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Same		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Patrick McCabe (IRDA)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

DHIL
I&W Data Base
Scenario
Remote Terminal Emulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objective of this effort was to identify the data base management system that seems to best support the Indications and Warning analyst using a given hardware and operating system configuration. Hardware and operating system constraints were to utilize a Digital Equipment Corporation PDP 11/70 computer running the IAS operating system. This configuration is the DOD standard intelligence support processor.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A precise description of an Indications and Warning data base was developed from scenarios constructed by observing analysts at DIA, SAC, MAC, and ADCOM. The description was then used to construct a simulated data base. Scripts were developed from each scenario to generate activity against the data base. Software was developed to allow execution of repeatable tests, each with psuedo-randomly variable conditions.

Important developments of this effort included; the precise characterization of an I & W Data Base, the DHIL language, used to describe the scenarios and the data base, the set of analyst scenarios, which accurately describe the functions and activities of intelligence analysts, and a hardware/software configuration usable for evaluating the performance of data base management hardware and software in the future.

SUMMARY

Research was conducted involving Data Base Management System (DBMS) characteristics for Indications and Warning (I&W). Actions and requirements of military intelligence analysts were reviewed, and a set of six scenarios were prepared to serve as a basis for computer simulation studies.

The Database Highorder Interactive Language (DHIL) was developed for precise representation of the scenarios. A preliminary review of DBMSs identified three systems for further testing: ADABAS-M, ORACLE, and SEED. Benchmark tests of the systems were performed, using DEC PDP-11/70 equipment under the IAS operating system and employing Remote Terminal Emulation Techniques. Difficulties with all three systems were encountered, and none may be entirely suitable for the application.

Keywords:

- Data Base Management Systems
- Military Intelligence Analysis
- Indications and Warnings
- Computer Simulation
- Remote Terminal Emulation

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF APPENDICES

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION AND SUMMARY

1.1 Purpose of the Final Technical Report

This Final Technical Report for the Indications and Warning Data Base Management System Analysis (Contract No. F30602-80-C-0286) is written to provide:

a. A summary of tasks undertaken and completed during the course of the project.

b. Results of experiments and tests performed.

c. Recommendations resulting from experiments, tests, and other evaluation procedures.

The objective of the project was to conduct research involving Data Base Management System (DBMS) characteristics for Indications and Warning (I&W). The I&W analysts within the USAF currently do not have a comprehensive data base of relevant information. Data are stored randomly in various forms and formats. No DBMS aids are employed to assist in storage, retrieval, and analysis. The thrust of this project has been a first effort to develop I&W DBMS characteristics.

The analysis was performed by Measurement Concept Corporation (Mc^2), with the assistance of Betac Corporation under subcontract. Five major tasks were performed:

- o A review of various types of intelligence analysis at SAC, ADCOM, and DIA to develop the characteristics of the functions and present operations. In addition, an I&W data base to support the analyst was designed and a simulation of the data base was developed.

- A study of available state-of-the-art DBMS technology which could be recommended as a testbed DBMS. Three DBMS packages were selected for benchmark testing.

- Development of testbed software.

- Benchmark testing of the selected DBMSs against such evaluation factors as data base creation, update, storage and retrieval.

- A recommendation of a DBMS with associated hardware to solve the I&W problem.

1.2 Summary of Results

Three Data Base Management Systems (DBMSs) were selected for extensive testing:

- ADABAS-M, an indexed system

- ORACLE, a relational system

- SEED, a network system using CODASYL conventions

Of the three DBMSs tested during this project, ADABAS-M is the undisputed, in fact, unchallenged winner by measures of efficiency. It loaded faster, retrieved faster (with the exception of a single test), and it updated faster. Furthermore, the areas recommended for improvement appear to be areas in which improvements could be accomplished with relative ease.

Although ORACLE did not compare favorably to ADABAS-M in performance measures, it is by nature at somewhat of a disadvantage. ORACLE provides more functional power in its non-procedural language and it supports a more general and basic data model than does ADABAS-M. In computer systems, enhanced functional power is almost always achieved at the expense of additional resource utilization.

SEED could not be tested as thoroughly as desired. To do so, would have consumed a disproportionate amount of project labor resources. The effort required to develop a responsive data base under the CODASYL model of data supported by SEED is truly enormous. The model is potentially conducive to efficient processing in restricted, stable, well defined enterprises. In such cases it would be worth the large design effort. Results during this project showed that; without custom designed input, SEED is prohibitively slow in the initial data base load. A data base design created with one type of query in mind will not likely support other t s of queries. The flexibility and expandability of CODASYL data base is severely limited.

Whatever their relative merits may be, examination of the results of these tests shows that none of the three DBMSs have performance characteristics required to support I&W systems of the near future. Even the most efficient of the three, ADABAS-M, cannot be expected to support more than approximately 25 typical I&W analyst requests per minute. The addition of data intensive inference and decision aid applications, supporting numerous analysts, will require greater data throughput than can be supplied by any of the three DBMSs in question.

ADABAS-M would support typical I&W centers of today. ORACLE, because of its much greater logical power and ease of use, might be preferred for smaller I&W centers (up to 5 analysts), but it cannot adequately support the larger ones.

The I&W centers of the near future will support more analysts, will support more comprehensive data bases, and will include sophisticated, data intensive, inference logic applications and decision aids. To achieve these goals, conventional computer systems will not suffice. What will be needed is special purpose machinery devoted to the data base task and serving the host as a peripheral. This machinery must employ parallel processing (eventually large scale), and some form of multiple channel data streaming (to avoid mechanical penalties) or multi-billion byte non-mechanical memories, preferably also with multiple channel access.

Such a device is currently called a "data base machine". Other devices which achieve some, but not all, of the characteristics of a data base machine are so termed. Some are currently commercially available, and some are in development. The IDM-500 is a good example of currently available data base machine technology. In full configuration, it is advertised as increasing data access speeds by a factor of 5 to 10. It should be benchmarked against the same scenarios performed during this project to assess its capability to support I&W analysis requirements for the near term (∿ 5 years). The IDM may be viewed as a relatively easily implemented, cost effective stop-gap measure to hold the line until true "multiple instruction, multiple data (MIMD)" machines are realized. An RADC sponsored MIMD machine, the Gaertner G-471, should be benchmarked against the scenarios used in this project to determine its capability to provide data base support in the mid-to-long term (5-10 years).

Several caveats are noted in the body of this report, of which the following are most signficant:

o In accordance with the terms of the contract, a DEC PDP-11/70 minicomputer under the IAS operating system, with two RP06 disk drives, was used for all experimentation. A more powerful central processor, a different operating system, and additional storage capacity would greatly improve performance characteristics for all systems tested.

o Problems with the implementation of ADABAS-M have led the vendor to withdraw the system from further sale. When and if this product is substantially modified and returned to the market, it should be reconsidered for possible recommendation.

o Errors and limitations in the SEED implementation may reflect the need to modify this system for operation under IAS. Performance under another operating system could be considerably more reliable.

- The version of ORACLE used for testing was 2.3. During the course of the project, a new version, ORACLE 3.0, was introduced but not tested. If the vendor's claims for improved performance are borne out in practice, the recommendation of ORACLE as the basis for an I&W analysis system would be strengthened.

- In accordance with the emphasis in the SOW on the relational data model, some bias toward this model existed throughout the design of the data base. If this emphasis were not justified, there may have been an unfair bias against SEED, which used a distinctively different data model.

- In particular, for applications in which the structure of the data base does not change frequently, and where the emphasis is placed on simple, routine queries, the greater efficiency of SEED could make it the system of choice.

Development of a testbed system for evaluation of selected DBMSs has provided RADC with an extensive hardware/software configuration for further testing and evaluation of tools and materials for the support of intelligence analysis. Among the products developed and delivered to RADC are the following:

- Scenarios representing detailed activities of intelligence analysts in their use of a projected DBMS.

- Comprehensive specifications of an I&W data base to support the scenarios.

- A simulated I&W data base for use in tests and cvaluation.

- Detailed specifications for the Database Highorder Interactive Language (DHIL) for representing I&W scenarios.

- Support programs to assist in translating from DHIL into applications programs for the DBMSs.

- Control programs for scheduling and monitoring performance of the scenarios.

- A complete Libyan Crisis scenario for research into analyst performance.

- Software to generate maps and other displays for use in demonstrations of system operation.

1.3 Project References

- Statement of Work for I&W Data Base Management Systems Analysis, PR No. I-0-4467, Rome Air Development Center, Griffiss Air Force Base, New York, 14 Jan. 1980; Amendment No. 1, 14 Aug. 1980.

- I&W Data Base Management System Analysis: Technical Proposal, 7 April 1980.

- *Working Paper: I&W Products, March 1981.*

- I&W Data Base: DHIL [Database Highorder Interactive Language], Script Syntax and Definition, 8 April 1981.

- Functional Analysis in Support of I&W Data Base Management System Analysis, 15 May 1981.

- DBMS Selection Study, June 1981.

- Conceptual I&W Data Base Specification, 15 June 1981.

- Initial DBMS Evaluation, 29 June 1981.

- Conceptual I&W Data Base Specification (modification of the 15 June report), 15 July 1981.

- Data Base Specification, July 1981, CDRL A002.

- General Purpose Monitor: Functional Description, Contract no. F30602-80-C-0277, July 1981.

- I&W Data Base Management System Analysis: Test Data Base Specification Plan, 4 August 1981.

- Experimental Application of Data Base Specification Procedures to the I&W Data Base, 21 August 1981.

- Application of Data Base Specification Procedures of the I&W Data Base Data Generation, 30 Sept. 1981.

- I&W Data Base Management Systems Analysis Project: Measurement Concept Corporation Configuration Management Plan, November 1981.

- I&W Data Base Management Systems Analysis Project: Benchmark Test Plan, January 1982.

- I&W DBMS Analysis Project: Interim Technical Report, March 1982, CDRL A003.

- DHIL Language/Syntax Diagrams, March 1982.

- Libyan Event Scenario & Data Base, 29 March 1982.

- Indications and Warnings Data Base Systems Analysis: Benchmark Test Specification, 31 March 1982.

- I&W DBMS Analysis: Artificial Intelligence Experiments, May 1982.

- I&W DBMS Analysis: Live Test Demonstration, 10 May 1982.

- I&W DBMS Analysis: Benchmark Test Results, December 1982.

1.4 Terms and Abbreviations

The following are terms, definitions, and acronyms used in this document and subject to interpretation by the user.

AI	- Artificial Intelligence
AOV	- Analysis of Variance
BMD	- Benchmark Driver
DBA	- Data Base Administrator
DBG	- Data Base Generator
DBMS	- Data Base Management System
EMT	- Emulated Trap
EWAMS	- Early Warning and Monitoring System
GPM	- General Purpose Monitor
HLI	- Host Language Interface
I&W	- Indications and Warnings
LTD	- Live Test and Demonstration
RTE	- Remote Terminal Emulation
SPM	- Special Purpose Monitor
SUT	- System Under Test
TIB	- Task Information Block
WEIS	- World Events/Interaction Series

2.0 DATA BASE MANAGEMENT SYSTEM SELECTION

This section provides a brief summary of the procedures used in selecting three Data Base Management Systems (DBMSs) for benchmark testing. Information is drawn from the project report "DBMS Selection Study," June 1981.

Two tasks supported this selection study:

- A functional analysis identifying I&W users and information products, which formed the basis for the scenarios presented in Section 3.0 of this report. Each of the scenarios was documented in flowchart form and was also described in the DHIL.

- A general survey of DBMSs suitable for implementation on DEC PDP-11/70 hardware under the IAS operating system. Eight candidate systems were identified, and vendor-supplied documentation, such as system descriptions and reference manuals, reports of previous performance and benchmark tests, and technical literature were reviewed. Nearly 180 DBMS features were considered in the survey.

Eight candidate systems were chosen. Their types and vendors are listed in Table 2-1.

The selection procedure involved three major steps:

- Define desired DBMS functions and their relative importance to I&W data base management needs.

- Rate each DBMS against the desired functions on a 0-10 scale, with 10 scoring highest.

- Total the scores based on the relative importance of the function.

DBMS	Vendor	Type
ADABAS-M	Software AG	Indexed
ORACLE	Relational Software	Relational
SARP-V	Bunker Ramo	Indexed
MADMAN	General Electric	Network/CODASYL
DBMS-11	DEC	Network/CODASYL
TOTAL	Cincom Systems	Network
SEED	International Data Base Systems	Network/CODASYL
REL*STOR	GTE	Relational

Table 2-1 Candidate DBMSs

On the basis of this formal selection process, the following DBMSs were designated for testing and evaluation:

- ADABAS-M, an indexed DBMS, developed for DEC minicomputer applications. It should be noted that ADABAS-M is a completely new product, not an adaptation of the older ADABAS system for IBM equipment.

- ORACLE, a relatively new relational DBMS.

- SEED, a network-type DBMS based on CODASYL conventions.

In addition to these three systems, three others were identified as candidates for further study:

- SARP-V

- REL*STOR

- A data base machine

SARP-V is currently widely used as an intelligence DBMS, and might be used as a baseline system. REL*STOR might have been of interest as an alternative to ORACLE when a relational system is under consideration. However, REL*STOR has since been removed from the market. A data base machine might be considered as the cost of hardware continues to decline, making a specialized machine cost-competitive with a software DBMS running on a general-purpose processor.

The three systems selected for comparative evaluation, then, were ADABAS-M, ORACLE, and SEED, representing indexed, relational, and network approaches, respectively.

3.0 SCENARIO DEVELOPMENT

This section briefly describes the Database Highorder Interactive Language (DHIL), developed by Betac Corporation and revised by Mc^2, for the precise, detailed representation of I&W analyst scenarios. It also describes the methods by which the scenarios were developed. A detailed description of the DHIL is provided in Appendix A of this report. English language versions of the scenarios are contained in Appendix B. A full presentation of the scenarios, written in DHIL, is contained in Appendix C.

The DHIL is a specialized language for scenario representation. It incorporates calls to the DBMS, control structures, and provisions for "think times," or designated pauses in scenario execution. Use of the DHIL permits a precise, unambiguous specification of scenario elements, including control elements (such as loops or repeated actions, conditional actions, and randomly determined actions). The DHIL specifications can then be translated into the specific formats required by the DBMSs under test. In general, these were FORTRAN programs calling on specialized subroutines supplied by the vendors. Use of the DHIL for specification of the scenarios helped to ensure that all DBMSs would be tested against the same benchmark.

A "scenario" contains a list of commands that the simulated user sends to the system under test. The scenarios provide for "think times," which represent periods during which the user is not interacting with the system. The DHIL specifications also make it possible to define control structures, such as repeated actions or a return to a previous action.

To simulate the use of the system by several analysts, replications of some of the scenarios were used. The use of multiple copies of the same scenario would have created an unrealistic simulation if all of the scenarios were accessing the same records at the same time. To prevent this situation, and to add to the realism of the simulation of multiple terminals, the system random number generator is used to provide a sequence of pseudo-random numbers as the basis for the selection of records to be accessed. The sequence of pseudo-random numbers is a function of the "seed," or starting value for the random number generator. Through the use of identical seeds, identical simulations could be generated for each of the DBMSs under test.

The scenarios provide an operational definition of the user's requirements. If a DBMS can execute a scenario, then it meets the minimum requirements of the users. The better it does on the scenario (in terms of time, resource requirements, costs, and other evaluation factors), the better it is for the intended application.

Scenarios were created by Betac Corporation, based on interviews with and observation of I&W analysts at ADCOM, MAC, SAC, and the NMIC. Analyst actions contained in the scenarios are felt to be typical of I&W analysts in normal and crisis operations. An additional scenario provides repeated updates of the data base, to test DBMS update capabilities.

The following six scenarios were developed:

- Watch
- I&W Analysis
- Collection Coordination
- Area Analysis
- Military System Analysis
- MAC Route Assessment

Three additional scenarios were designed by Betac but not fully developed:

- o Space Event Assessment
- o Missile Threat Assessment
- o ELINT Analysis

Scenarios were varied in the following ways:

o Pseudo-random numbers were used to simulate variations in performance of the analyst tasks. Since the pseudo-random numbers are functions of the "seed" (a number which is used to initiate a sequence of pseudo-random numbers), the same set of actions could be performed for each of the systems under test, simply by using the same seed for each test.

o Varied levels of alert (peace, crisis, war) may be simulated. These affect the rate at which actions are performed, primarily by varying the "think times."

o Varied loads were introduced, including variations in the number of terminals and in the number of scripts to be processed at each terminal.

Following receipt of the scenarios from Betac, Mc^2 personnel reviewed and revised them to remove inconsistencies and to eliminate incompatibilities with the data base designs. Mc^2 also prepared an "update scenario," consisting of repeated updates to the data base, to test the efficiency and reliability of this operation. The scenarios were then reviewed by RADC/IN, which reported that they accurately represented analyst activities.

The scenarios were next translated into "scripts," which are computer programs written in a language (such as FORTRAN) which is acceptable to a specific system under test. The scripts were executed under the control of the Benchmark Driver. Once an individual script began execution, it functioned as a separate task, issuing calls to the DBMS at specified intervals. "Think times," during which the emulated analyst was reading messages and performing other tasks, were represented by wait times within the scripts.

Execution of the scripts was monitored by the General Purpose Monitor (GPM), which is described in Section 8.0. The GPM records data concerning resource utilization and other performance factors.

The scenarios are a major resource for testing and evaluating DBMSs. They represent analyst activities at a level of detail which has not been otherwise obtainable. With their accompanying data base, they will be valuable for testing and demonstrating other DBMSs and components of an I&W analyst facility, within the context of an Intelligence Systems Laboratory.

4.0 TASK INFORMATION BLOCKS (TIBS)

Task Information Blocks (TIBs) are specifications of individual files to be created to support the scenarios. The TIBs were designed by Betac, and were reviewed and modified by Mc^2 to eliminate inconsistencies with the scenarios.

The TIBs used for the simulations are described in Appendix D of this report. Since the TIBs specify the contents of an I&W data base of sufficient size and complexity to support the scenarios, they represent one of the contributions of this project to future development of an I&W data management system.

The following TIBs were specified:

A001. Message Processing Input/Output (based on GENSER message traffic)

B001. Event Records

C001. Order of Battle

D001. Weapons/Equipment Summary

D002. Personalities

E001. Country Summary Information

E002. Installation Data

F001. Indicator Lists

G001. Collection Requests

H001. Friendly Mission Information

I001. Air Route Mission Data

I002. Air Route Segment Data

J001. Weather Data

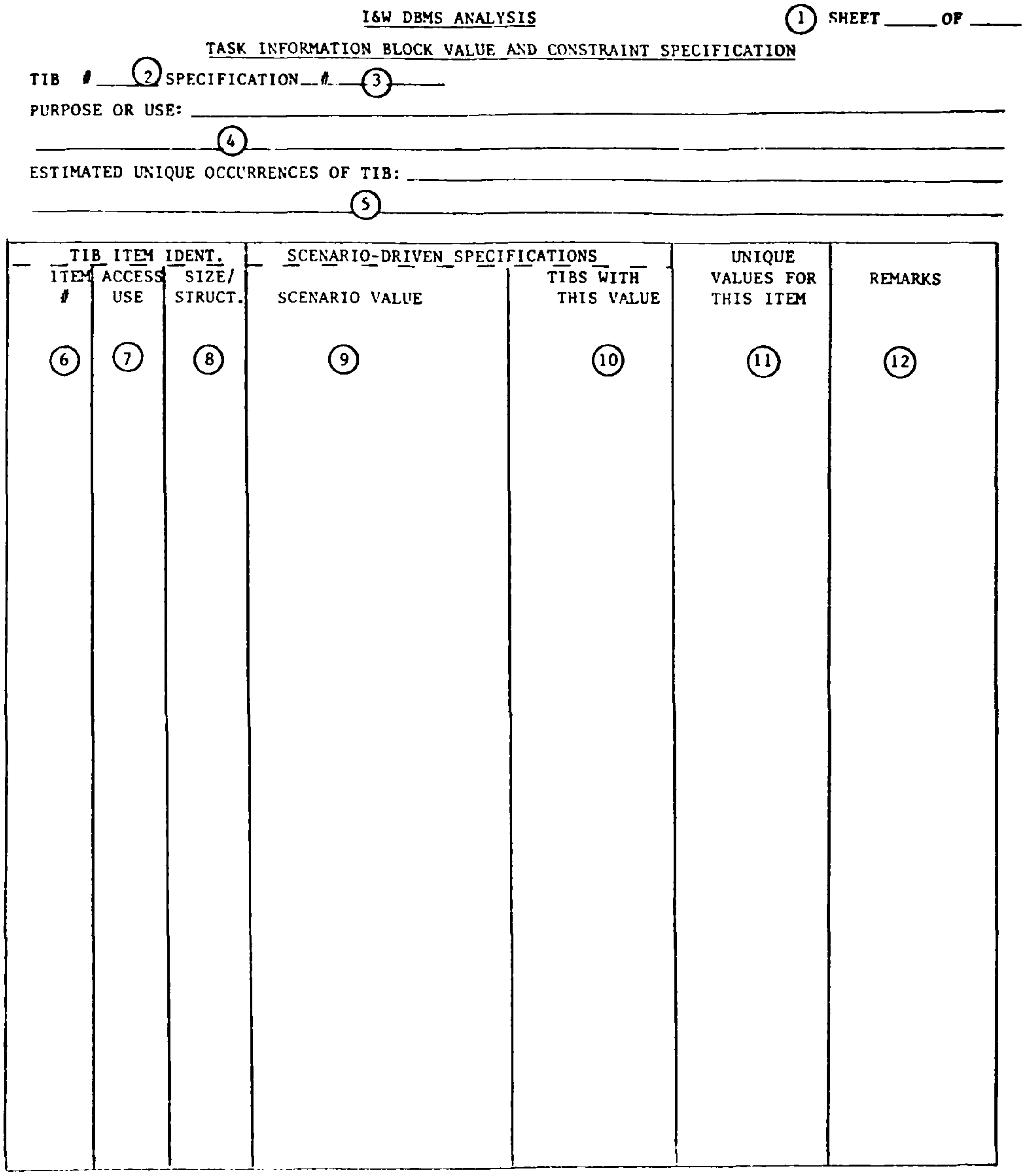
A Data Base Generator (DBG) was used in conjunction with the TIBs to generate flat (unstructured) files, which provided a simulated I&W data base which was not specifically tailored for any one of the DBMSs to be tested. The TIBs specified the range and distribution of data within each field. The DBG randomly selected values to be placed in the data base, within the range and with the distribution specified by the TIBs. Thus, while the data are composed simply of random values, they have the statistical characteristics of a realistic I&W data base. This approach permitted the construction of a data base which would provide statistically valid tests of the DBMSs without the high labor costs and security considerations involved in implementing an I&W data base with real data.

Figure 4-1 represents a blank TIB form. Numbers appearing on the form correspond to numbers in the descriptions given below.

1. Sheet numbers refer to those required for the complete specification of one TIB.

2. The TIB Number is the identification code.

3. The Specification Number identifies the particular specification statement.



I&W DBMS ANALYSIS ① SHEET ____ OF ____

TASK INFORMATION BLOCK VALUE AND CONSTRAINT SPECIFICATION

TIB # __②__ SPECIFICATION # __③__

PURPOSE OR USE: ______________________________

__________④__________

ESTIMATED UNIQUE OCCURRENCES OF TIB: ____________________

__________⑤__________

TIB ITEM IDENT.			SCENARIO-DRIVEN SPECIFICATIONS		UNIQUE VALUES FOR THIS ITEM	REMARKS
ITEM #	ACCESS USE	SIZE/ STRUCT.	SCENARIO VALUE	TIBS WITH THIS VALUE		
⑥	⑦	⑧	⑨	⑩	⑪	⑫

Figure 4-1 Form Used for Specification of Task Information Block Values

4. The purpose or use is an explanation of the context within which the TIB occurs, and the reason for its occurrence in the form shown.

5. The number of unique occurrences of the TIB refers to the total count of all TIB occurrences which are associated with the given specification.

6. The TIB item number.

7. The access use:

   "G" Grouping item.

   "E" Entry item or key. (Occasionally, "E1" and "E2" are used. These refer to entry keys which are a combination of the items marked, taken in the order numbered.)

8. Size and structure information.

9. The specific value to be used in this specification.

10. The number of occurrences of the TIB in which the particular value stated will occur.

11. The total number of different and unique values which the given TIB item can be expected to take, within the context of the TIB's functional use. This information is given only for entry and key items.

12. Remarks cover any extra information required for the specification. Of particular importance here is the statement of the number of occurrences of a subblock to be expected for one occurrence of a header block.

5.0 DATA BASE GENERATION

The procedures used for the test data base specification were developed following recommendations contained in the Betac report, "Application of Data Base Specification Procedures of the I&W Data Base Generation," 30 September 1981. The procedures used for determination of the values and distribution of values in the I&W data base were as follows:

- For each item in the TIBs, estimate its expected value, range of values, and distribution of values, based on the Betac report.

- Taking into account the scenarios to be implemented in the Benchmark Tests, select specific values to support the activities described in the scenarios.

Two types of files were built, using the line editor, as input to the Data Base Generator (DBG) program:

- *.TIB files. These contain the name of the TIB, number of records, number of blocks, and specific values of each item.

- *.TAB files. These contain the number of occurrences of the miscellaneous table driven constants.

The *.TIB and *.TAB files were used as input to the DBG program to produce flat-file representations (i.e. unstructured files) of the data base.

The exact format of the files to be used by the loader was different for each DBMS. For ADABAS-M a conversion program, ADACNV, was written to convert output files produced by the DBG into the required format for the ADABAS-M loader.

Values generated by the DBG were then checked to ensure that the generated data actually met the TIB specifications. A number of errors were detected and corrected through the use of small utility programs. In general, these errors were failures to generate special values required by some of the scenarios. (Errors were detected and corrected in files B1, C1, D1, D2, E1, and E2.) These programs used the flat files produced by the DBG program as input, and corrected or modified the faulty fields before they were passed to ADACON, the conversion program for reformatting the data into the format desired for the ADABAS-M loader.

As pointed out earlier, a number of small corrector programs were cycled against the intermediate versions of the data base before it was loaded for ADABAS-M. It was deemed more resource-effective, in both time and manpower, to download the data base from ADABAS-M and reformat/convert the files into the formats required by the high-speed load facilities of ORACLE and SEED. The advantage to this approach was twofold:

o Each item in the data base for each DBMS has exactly the same value. Any error in value present in the data base for ADABAS-M will also be present for ORACLE and SEED.

o The approach was more cost effective. Due to the size and complexity of the files in the data base, a number of long-running jobs utilizing some very large intermediate files were necessary to build each file. Some of these large intermediate files could not reside on the same disk pack, necessitating numerous mounts and dismounts of disk packs during the course of building the data base.

6.0 WEIS/EWAMS DATA BASE

This section describes the Early Warning and Monitoring System (EWAMS) data base, which is derived from the World Events/Interaction Series (WEIS) data base. The WEIS data base originally used the New York Times as its sole source but has since added The Times of London and the Los Angeles Times as additional sources of data. The New York Times is the principal source, and the other two papers are used as supplementary sources, to check reliability and validity [McClelland, Charles A., "D-Files for Monitoring and Forecasting Threat and Dangerous Problems Abroad." Paper presented at the Annual Meeting of the International Studies Association, Washington DC, March, 1978]. EWAMS also uses The Guardian of Manchester as a source.

Events from these papers are put into one of 63 different categories. In order to facilitate uniform coding, these 63 categories are ordered and subdivided into 22 groups. Each of the 22 groups is headed by a cue-word. The cue-words are not intended to have their exact common-language meanings, but are technical mnemonic devices only. Examples of the cue-words are: YIELD, COMMENT, CONSULT, and APPROVE. Under each of the cue-words are listed one or more categories, each of which is given a numerical code. For example, looking under the third cue-word, CONSULT, one would find:

3. CONSULT

031 Meet with at neutral site; or send note;
stay in same place
032 Visit; go to; leave country
033 Receive visit; host

(See Table 6-1 for a complete listing of the categories and their codes. See also Table 6-2 for a listing of the Conflict Area Codes, and Table 6-3 for a listing of the Actors Codes.)

In addition to the code for 63 actions, there is a numerical code for the participants in the interactions. The originator of each action is called an "actor," and the recipient is termed a "target." These actors and targets include both independent nations and international organizations and groups, such as NATO and the European Common Market. For example, 002 is the code for the United States, 395 for Iceland, and 396 for NATO. With this coding, event interactions may be concisely represented in the data base. For example, if the United States deployed naval vessels to protect American fishing boats near Icelandic waters from harassment, the event would be encoded:

002 223 395

where 223 is the code for "military engagement." Actually, the encoded message in the WEIS data base would also include code for date, coder, collection number, and item identification number. It would look like this:

83 07 18 002 223 395   00 25   990001

The numbers give the year, month, day, actor, event, target, source, coder, and an item identification number. It is important to note that the encoded message would be identical if the United States had launched a massive nuclear attack on Iceland. The intensity and the context of the interaction are ignored in the WEIS data base. It is assumed that the number of messages will provide a rough measure of the intensity of the interaction.

NUMBER	ALPHA	ACTOR
700	AFG	Afghanistan
339	ALB	Albania
615	ALG	Algeria
232	AND	Andorra
160	ARG	Argentina
900	AUL	Australia
305	AUS	Austria
053	BAR	Barbados
211	BEL	Belguim
266	EBE	Berlin/East
267	WBE	Berlin/West
145	BOL	Boliva
571	BOT	Botswana
140	BRA	Brazil
355	BUL	Bulgaria
775	BUR	Burma
516	BUI	Burundi
811	CAM	Cambodia
471	CAO	Cameroun
020	CAN	Canada
482	CEN	Central African Rep.
780	CEY	Ceylon
483	CHA	Chad
155	CHL	Chile
710	CHN	China, People's Rep.
713	CHT	China, Rep. of
100	COL	Columbia
484	CON	Congo (Brazzaville)
490	COP	Congo (Kinshasa)
094	COS	Costa Rica
040	CUB	Cuba
352	CYP	Cyprus
315	CZE	Czechoslovakia
434	DAH	Dahomey
390	DEN	Denmark
042	DOM	Dominican Rep.
130	ECU	Equador
092	ELS	El Salvador
440	GUE	Equitorial Guinea (incl. Fernando Po)
530	ETH	Ethiopia
375	FIN	Finland
220	FRN	France
481	GAB	Gabon
420	GAM	Gambia
265	GME	Germany/Dem.Rep
255	GMW	Germany/Fed.Rep
452	GHA	Ghana
350	GRC	Greece
090	GUA	Guatemala
438	GUI	Guinea
110	GUY	Guyana
041	HAI	Haiti
091	HON	Honduras
310	HUN	Hungary
720	HOK	Hong Kong
395	ICE	Iceland
750	IND	India
850	INS	Indonesia
630	IRN	Iran
645	IRQ	Iraq
205	IRE	Ireland
666	ISR	Israel
325	ITA	Italy
437	IVO	Ivory Coast
051	JAM	Jamaica
740	JAP	Japan
663	JOR	Jordan
501	KEN	Kenya
731	KON	Korea/North
732	KOS	Korea/South
690	KUW	Kuwait
812	LAO	Laos
660	LEB	Lebanon
570	LES	Lesotho
450	LBR	Liberia
620	LBY	Libya
223	LIC	Liechtenstein
212	LUX	Luxemburg

Table 6-1 Alphabetical Listing of WEIS Actors (page 1 of 2)

NUMBER	ALPHA	ACTOR
721	MAC	Macao
580	MAG	Malagasy
553	MAW	Malawi
820	MAL	Malaysia
781	MAD	Maldive
432	MLI	Mali
338	MLT	Malta
590	MAR	Mauritius
435	MAU	Mauritania
070	MEX	Mexico
221	MOC	Monaco
712	MON	Mongolia
600	MOR	Morocco
698	MOM	Muscat and Oman
921	NAU	Nauru
790	NEP	Nepal
210	NTH	Netherlands
920	NEW	New Zealand
093	NIC	Nicaragua
436	NIR	Niger
475	NIG	Nigeria
385	NOR	Norway
770	PAK	Pakistan
095	PAN	Panama
150	PAR	Paraguay
135	PER	Peru
840	PHI	Philippines
290	POL	Poland
235	POR	Portugal
552	RHO	Rhodesia
360	RUM	Rumania
517	RWA	Rwanda
331	SAN	San Marino
670	SAU	Saudi Arabia
433	SEN	Senegal
451	SIE	Sierra Leone
830	SIN	Singapore
520	SOM	Somalia
560	SAF	South Africa
230	SPN	Spain
625	SUD	Sudan
572	SWA	Swaziland
380	SWD	Sweden
225	SWZ	Switzerland
652	SYR	Syria
510	TAZ	Tanzania
800	TAI	Thailand
461	TOG	Togo
052	TRI	Trinidad-Tobago
616	TUN	Tunisia
640	TUR	Turkey
500	UGA	Uganda
365	USR	USSR
651	UAR	UAR (Egypt)
200	UNK	United Kingdom
002	USA	USA
439	UPP	Upper Volta
165	URU	Uruguay
328	VAT	Vatican
101	VEN	Venezuala
817	VTS	Vietnam/South
816	VTN	Vietnam/North
990	WSM	Western Samoa
678	YEM	Yemen
681	SYE	Yemen/South
345	YUG	Yugoslavia
551	ZAM	Zambia
Non-Governmental Actors		
198	AFP	Alliance for Progress
699	ARL	Arab League
476	BIA	Biafra
397	EEC	Common Market
398	EFT	EFTA
396	NAT	NATO
199	OAS	OAS
599	OAU	OAU
697	PLO	Arab Commando groups
813	LAP	Pathet Lao
992	SEA	SEATO
818	VCG	Vietcong and NLF
394	WAR	Warsaw Pact
399	UNO	Any intl. org.
998	MLG	Any multilateral group
999	NSC	Not stated, unidentified target

Table 6-1 Alphabetical Listing of WEIS Actors (page 2 of 2)

1. YIELD

011 Surrender, yield to order, submit to arrest, etc.
012 Yield position; retreat; evacuate
013 Admit wrongdoing; retract statement

2. COMMENT

021 Explicit decline to comment
022 Comment on situation--pessimistic
023 Comment on situation--neutral
024 Comment on situation--optimistic
025 Explain policy or future position

3. CONSULT

031 Meet with at neutral site; or send note; stay in same place
032 Visit; go to; leave country
033 Receive visit; host

4. APPROVE

041 Praise, hail, applaud, condolences, ceremonial greetings, thanks
042 Endorse others policy or position, give verbal support

5. PROMISE

051 Promise own policy support
052 Promise material support
053 Promise other future support action
054 Assure; reassure

6. GRANT

061 Express regret; apologize
062 Give state invitation
063 Grant asylum
064 Grant privilege, diplomatic recognition; de facto relations, etc.
065 Suspend negative sanctions; truce
066 Release and/or return persons or property

7. REWARD

071 Extend economic aid (gift and/or loan)
072 Extend military assistance; joint military exercises
073 Give other assistance

8. AGREE

081 Make substantive agreement
082 Agree to future action; agree to meet, negotiate; accept state invitation

Table 6-2 Listings of WEIS Event Interaction Codes (page 1 of 3)

9.	REQUEST
091	Ask for information
092	Ask for policy assistance; seek
093	Ask for material assistance
094	Request action; call for; ask for asylum
095	Entreat; plead for; appeal to; help
10.	PROPOSE
101	Offer proposal
102	Urge or suggest action policy
11.	REJECT
111	Turn down proposal; reject protest demand, threat, etc.
112	Urge or suggest action or policy
12.	ACCUSE
121	Charge; criticize; blame; disapprove
122	Denounce; denigrate; abuse; condemn
13.	PROTEST
131	Make complaint (not formal)
132	Make formal complaint or protest
14.	DENY
141	Deny an accusation
142	Deny an attributed policy, action, role, or position
15.	DEMAND
150	Issue order or command, insist; demand compliance, etc.
16.	WARN
17.	THREATEN
171	Threat without specific negative sanctions
172	Threat with specific non-military negative sanctions
173	Threat with force specified
174	Ultimatum; threat with negative sanctions and time limit specified
18.	DEMONSTRATE
181	Non-military demonstration; walk out on; boycott
182	Armed force mobilization, exercise, and/or display

Table 6-2 Listings of WEIS Event Interaction Codes (page 2 of 3)

19. REDUCE RELATIONSHIP (As Negative Sanction)

191 Cancel or postpone planned event
192 Reduce routine international activity; recall officials, etc.
193 Reduce or suspend aid or assistance
194 Halt negotiations
195 Break diplomatic relations

20. EXPEL

201 Order personnel out of country; deport
202 Expel organization or group

21. SEIZE

211 Seize position or possessions
212 Detain or arrest person(s)

22. FORCE

221 Non-injury destructive act; bomb with no one hurt
222 Non-military injury-destruction
223 Military engagement

Table 6-2 Listings of WEIS Event Interaction Codes (Page 3 of 3)

CODE	CONFLICT ARENA
010	Arab-Israeli conflict (general)
013	1967 war (all Mideast events during 1967)
020	Vietnam talks in Paris
025	1968 Vietnam talks in Paris
027	Military engagements as of October 1969 (Vietnam)
030	Rhodesian independence
040	Berlin conflict
050	Sino-Soviet conflicts
060	Indonesia-Malaysia disputes
070	India-China conflicts
080	USA-China conflicts
090	India-Pakistan disputes
100	Cyprus
110	Korean conflicts
120	France-NATO dispute
130	West German-East Europe dispute
150	Dominican Republic
160	Red Guard activities
170	Czechoslovakia-Soviet Union
180	Biafra-Nigeria conflict
190	Strategic Arms Limitation Talks
200	Non-Government-Sanctioned Violence (e.g., kidnappings, hijackings, etc.)
210	Cambodian conflict

From C.A. McClelland, et al "The Management and Analysis of International Event Data: A Computerized System for Monitoring and Projecting Event Flows", 1971.

Table 6-3 Listings of Conflict Arena Codes

In addition to actor, event, and target codes, there are arena codes that denote areas that are of particular interest. This feature allows the user of the data base to extract all data pertinent to a particular international occurrence, e.g. the Vietnam conflict, or the India-China conflicts. This code would follow the target code. Thus one would encode the event of India's protesting China's deployment of troops near India's border as:

71 07 18 750 313 710 070   00 25   990002

where the first three entries are the date, 750 is the code for India, 313 is the code for an informal complaint, 710 is the code of the People's Republic of China, and 070 is the code for India-China conflicts. [C. A. McClelland et al., "The Management and Analysis of International Event Data: A Computerized System for Monitoring and Projecting Event Flows," World Event/Interaction Survey Technical Report, Los Angeles: University of Southern California, Sept. 1971.]

For implementation and demonstration during this project, a subset of the records in the EWAMS data base were expanded to include latitude and longitude of each event. In this way, the location of the event could be used in a map display of world events, and could be used to retrieve messages in a selected area.

The WEIS data base consists of two parts. In addition to the coded portion of the data base discussed above, there is another section of the WEIS data base that contains a brief English language summary of each coded interaction. These "messages" were also used in demonstrations of the DBMSs.

7.0 LIBYAN SCENARIO DEVELOPMENT

A detailed scenario describing events related to the Libyan crisis of 1981 was produced by Betac. Only unclassified data were used in generating the scenario and supporting material. The source of most of this information was the daily Foreign Broadcast Information Service (FBIS) reports published by the U.S. Government. These reports are issued in editions covering each of the world's major regions, and containing extracts from broadcasts of radio stations and from newspapers of the regions. This test exercise of the I&W DBMS uses the Middle East and Africa daily reports. The entries for Libya and those concerned with Libyan affairs and events listed under other nations in the region from 1 July 1981 to 1 October 1981 were reviewed for possible correlation with items contained in an ad hoc set of Middle East indicators formerly used by DIA.

The FBIS information was used as the basic data source for several reasons. First, it provides an unclassified ongoing source of data for a specific nation or geographic area. This information is of greater breadth and volume, and of more detail for individual countries, than standard newspapers or journals published in other English language sources. In addition, its natural focus tends to be upon articles reflecting I&W concerns. The FBIS data, then, provide an equivalent to the stream of messages with which an intelligence analyst might have to deal while manning a position in an I&W center.

In the suggested exercise of the data base, the analyst would be attempting to determine the present level of threat to various American activities in the region that occur within the operational area of a given nation's armed forces and weapon systems. In this case the specified nation is Libya. The analyst is also attempting to determine any changes in threat level which new events relating to Libya may indicate. The following summary provides an outline of the material contained in the Libyan scenario.

The scenario represents the perception of the threat presented by Libya to U.S. forces, citizens, and interests, based on recent Near Eastern and Middle Eastern events. The analyst would presumably have knowledge of the U.S. Navy maneuvers taking place in the Gulf of Sidra during August 1981. The first alert message, or one that would cause the analyst to feel increased concern over U.S. activities in the area, is a warning statement from the Libyan government threatening the U.S. Navy exercise. This is followed by a report of the actual air clash, with later reports of reactions to the clash. These include condemnation of the U.S. action in the press by several Arab countries, including some at least nominally friendly to the U.S. Some of these articles call for punishment of the U.S. and a unified stand in support of Libya.

In terms of further threats to the U.S. or to U.S. activities in the area, the analyst would note that Colonel Qaddhafi is again threatening to join the Warsaw Pact and possibly to allow the Soviets to have basing rights in Libya. At the same time, Qaddhafi also threatens to attack NATO bases in Southern Europe, should the U.S. again "attack" Libyan forces. The analyst also finds an item stating that the Libyans have Soviet made medium-range missiles that could reach Southern European NATO bases from launch sites in Libya.

In several reports which are possibly related, the analyst finds that there have been defections of Libyan diplomats in posts around the world. If they can be used an as indication of increased dissention within the Libyan government over policies of the ruling group, one might surmise that incidents such as the 19 August air battle could have been precipitated by the Libyan government in order to create a public perception of an external threat, thus aiding in quelling dissension by appealing to a need for national unity. This may also mean that further incidents can be expected if the desired effect is not produced by the initial incident or if the feeling of national unity is to be maintained at a high pitch over an extended period of time.

Overall, then, the analyst sees a violent incident occurring quite suddenly, followed by a strong reaction from the Libyan government. Whether or not the Libyan government deliberately initiated the incident, it appears from their reaction that they were genuinely frightened by the possibility of further incidents. The result was what could be the strongest anti-U.S. threats by the Libyans to date. The nature of these threats -- especially those directed toward U.S. nuclear weapons stockpiled at Southern European NATO bases -- and the degree of support for Libyan retaliation exhibited by some of the more moderate Arab states may cause the analyst to focus his efforts on Libyan threats as very serious short term or intermediate term disruptive factors in the Near East and Mediterranean area.

In this scenario, the analyst performs a sequence of operations that could be partially supported by a DBMS:

- Background information is retrieved concerning U.S. Navy Maneuvers in the Gulf of Sidra.

- A statement from the Libyan government concerning the maneuvers is perceived as a threat. In other words, the statement must be interpreted in such a way as to indicate its relevance for the analyst.

- A report concerning the air clash must be correlated with reports of reactions to the clash. The event gains in signficance as these additional reports are clustered with the original report.

- The fact that some of the condemnations come from friendly nations is more signficant that the routine condemnations from hostile nations.

- Qaddhafi's repeated threats to join the Warsaw Pact must be correlated with the developing picture of the situation. Historical information concerning earlier threats of this type should be retrieved automatically, to supplement current information.

- The system should have some means of relating the potential crisis to the earlier defections of Libyan diplomats. It should be possible to see that these defections can be taken as a possible cause of the violent reaction to the U.S. activities, and to hypothesize that the Libyan government might have initiated the incident.

- The system should be capable not only of evaluating but of characterizing the threat in detail, and of providing an explanation of the reasoning that lies behind its evaluation.

By isolating specific requirements such as these, the Libyan scenario provides the basis for recommending further enhancements and extensions to the I&W DBMS.

8.0 SCHEMA DEFINITIONS

Schema definitions were prepared in the forms required by each of the three DBMSs. In theory, it should have been possible to load the flat files directly onto disk, in the specified formats, using the bulk loader facilities of the DBMSs. In actuality, the process was more complex. For ADABAS-M, it was necessary to prepare special programs to transform the data base into formats which were acceptable to the ADABAS-M loader. The transformed data were also used with the ORACLE loader. Preparation of schema definitions for SEED proved to be considerably more difficult than for the other DBMSs, and only a subset were loaded. In general, the size of the data base required to support the I&W scenarios was several times larger than the capacity of the two RP06 disk drives available, and subsets were used for the actual testing.

A listing of the files actually implemented is contained in Section 12.0 of this report. Detailed schema definitions are being supplied with program documentation for this project.

9.0 GENERAL PURPOSE MONITOR

Remote Terminal Emulation (RTE) tests were monitored by the General Purpose Monitor (GPM), a software tool developed by Mc^2 under RADC Contract No. F30602-80-C-0277. The GPM will be described in this section.

The GPM is an evaluation tool which measures system activities and produces summary reports for use in assessing system performance. Two goals in the design and implementation of the GPM were:

- Low system artifact (that is, the GPM should require a minimum of system resources for its own operation).
- Zero system endangerment (that is, operation of the GPM must not cause failures of the system under test).

In meeting these objectives, the GPM was designed to operate externally to the system under test, observing its behavior from the outside. This approach does not require user modification of the program code, through the insertion of probes, counters, flags, and the like. Instead, the GPM operates by modifying Operating System code to support monitoring of various conditions at appropriate times during system activities. The types of data collected depend upon the options selected by the user.

In addition to response time, which can be measured indirectly by the GPM, the monitor has the ability to collect data relating to the following activities:

- Emulated Traps (EMTs) executed by the system
- Send/Receive Directives
- Queue Inputs/Outputs (QIOs) by task and devices

- Task Execution Directives
- Exit and Suspend Directives
- Trap, Informational, and Status Directives
- I/O to files
- Block Counts of data to and from devices
- Null Task times
- Operating System Service time
- Task statistics and idle time
- Node Pool use
- Disk Arm Movement

The GPM is described in more detail in Appendix E.

10.0 GENERAL DESCRIPTION OF SYSTEMS TESTED

This section describes characteristics of the three systems under test, ADABAS-M, ORACLE, and SEED. The purpose of this section is to provide the reader with a fairly detailed picture of the systems, emphasizing those features that would be significant in a particular application.

The three systems represent three approaches to data base system design: indexed, relational, and network. Within these approaches, each of the systems contains features which make it more or less applicable to an individual installation. There is no system that is best at everything.

10.1 ADABAS-M Charateristics

As noted elsewhere in this report, ADABAS-M was developed by the vendor, Software AG, for use on DEC computer equipment. It is not an adaptation of the widely-used ADABAS system. Errors in the implementation have led the vendor to withdraw the ADABAS-M implementation from further sale. The system was nevertheless efficient and easily-used. It should certainly be considered for I&W applications when and if it returns to the market. Although the technology upon which it is based is some twenty years old, ADABAS-M can perform well for many users. The use of indexes provides very rapid access to data items when the keys are known, but non-keyed items can be retrieved only through the use of specially written programs, and retrieval is much slower.

Like the other systems, ADABAS-M provides a full range of update/load components: the ability to modify and edit individual data items within a selected record, the ability to add or delete individual records, and the ability to load large amounts of data from non-DBMS files into DBMS files through a bulk-load facility. In addition, ADABAS-M can use relational and Boolean operators to qualify records for modification or deletion. A host language has the ability to perform update/load functions from a programming language. Loaders are available for both initial and incremental load. A sort utility is included in the loader package.

All three systems provide a range of query and report facilities. ADABAS-M has the ability to perform query and report functions from a programming language. It can use relational and Boolean operators to qualify records and data items for retrieval. Functions for producing summaries are also provided. Applications programs must be used for testing non-indexed field values. ADABAS-M includes facilities for phonetic retrieval, which might be useful for some I&W files (e.g. biography files) where the exact spelling is not known.

ADABAS-M, like the other systems, includes a range of facilities for restart and recovery. It maintains an audit log which can record all transactions made to the data base. It is also capable of dumping the contents of the entire data base files onto removable storage (e.g. disk) and copying the contents of a dump into the data base files. Checkpoint, rollback, and rollforward facilities are provided. The system can reconstruct the data base from a known consistent state (checkpoint) after a system crash, by rolling committed transactions forward and uncommitted transactions backward. ADABAS-M provides utilities for transferring the disk log onto tape.

ADABAS-M provides facilities for protecting data security, including the ability to authorize selective access to a file, set, record, or data item. Read-only and write permissions, including add, delete, and update operations, can be specified.

A full range of concurrency controls are also provided. Multiple users may access the same collection of data, such as a file or set. Multi-user facilities are provided. Multi-threading, the ability to overlap service requests to secondary data storage devices, is also provided. ADABAS-M handles up to 128 threads.

All three systems provide facilities for data definition. ADABAS-M provides a schema capability to define the logical structure of the data base, as well as the physical organization, format, and validation criteria for the data. In addition, all systems provide for subschemas, or user views, corresponding to user applications. In ADABAS-M, indexes can be dynamically created or dropped.

All systems under test are capable of physically rearranging data for increased access efficiency and reduced data volume. ADABAS-M provides monitoring subfunctions, including the ability to determine and report on the status of the DBMS files, such as the number of records, the percent filled, etc., and the ability to determine and report on the current status and performance of the system, such as the number of users, number of I/Os, etc.

ADABAS-M compares favorably with the other systems in the size of the data base supported and its ability to accommodate large data bases and multi-volume files. Memory resources required by ADABAS-M are the smallest among the three systems tested, with only 32K words needed. In terms of system cost, ADABAS-M, at about $38,000, was somewhat lower in price than ORACLE ($48,000), but more expensive than SEED ($14,000).

ADABAS-M provides a full range of documentation, including user manual, operator manual, programming manual, and training materials. Training, provided on-site, included a competent company representative with appropriate instructional materials.

In terms of portability, ADABAS-M is the least portable of the three systems tested, since it was programmed in assembly language for DEC equipment. It is important to remember that ADABAS-M is not the same system as ADABAS, which is tailored to IBM equipment.

In Mc^{2}'s initial evaluation, summarized here, ADABAS-M received the highest score of all the systems rated. However, the scores of all three systems under test were quite similar. ADABAS-M was given high scores for its update/load facilities, large data base capacity, concurrency controls, and monitoring capability. It was given a somewhat lower rating for its lack of portability.

10.2 ORACLE Characteristics

ORACLE, distributed by Relational Software Incorporated, is a relatively new system exploiting the relational data model developed by E. F. Codd ["A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, 13:6 (June 1970)]. The system tested was ORACLE 2.3, which was replaced during 1982 by ORACLE 3.0. ORACLE 3.0 is said to be a more efficient implementation of the model.

ORACLE provides all standard update and load facilities, including the ability to modify and edit data, add and delete data items, perform a bulk load, and use relational and Boolean operators in specifying items for update and load. A host language interface permits the user to perform update and load functions from an application program.

ORACLE query and report facilities include the ability to perform functions from a host language, relational and Boolean operators, and built-in summary functions (for counts, averages, minima, and so on). The relational calculus used in ORACLE's query language allows multiple record types to be retrieved in a query (a relational join).

All standard restart and recovery features are provided, including an audit log, the ability to dump or copy entire data base files, and checkpointing with rollback and rollforward.

Security provisions for ORACLE were rated as satisfactory, having the ability to protect files against unauthorized access, and to authorize read-only or write permission. The data base administrator can authorize other users to set permissions.

Concurrency controls also include all standard features, supporting shared access to data files, multi-user access, and multi-threaded access. Data definition capabilities are similar to those of the other systems under test, including both schema and subschema definition facilities. New table columns and indexes can be added dynamically. The data base can be reorganized automatically for greater retrieval efficiency or better storage utilization. Multi-volume files are not supported.

Memory requirements for ORACLE are 80K words, making it the largest of the three systems under test. It is also the most expensive, with a cost (at the time of testing) estimated at about $48,000, some $10,000 more than ADABAS-M and $34,000 more than SEED. Cost was not heavily weighted in Mc^2's evaluation.

Documentation was comparable to that of the other systems tested, including user manual, operator manual, programming manual, and training materials. Although training was provided on-site, the vendor has discontinued on-site training in favor of training at the vendor's offices. ORACLE training was the least well-organized, with informal presentations of a series of viewgraphs by one of the vendor's representatives. The instructor depended heavily on questions from the participants. However, ORACLE is probably the easiest of the three systems to learn independently, without formal training sessions.

ORACLE is rated as fairly portable. It is written in the C programming language and is available in versions for IBM and DEC computer equipment.

Altogether, in Mc^2's initial evaluation, ORACLE scored nearly as well as ADABAS-M, and better than any other system considered. Its positive features were seen as its effective facilities for data definition, based on its relational data model, and its query/report functions. Negative factors included restrictions on the size of the data base, and its large memory requirement. The high cost may also be a negative factor in some applications.

10.3 SEED Characteristics

SEED is quite different in concept from ADABAS-M and ORACLE, emphasizing efficiency in data storage and retrieval, rather than flexibility and ease of use. Based on CODASYL recommendations, SEED uses a hierarchical network to represent data structures. The user must be aware of this network structure in order to use SEED effectively. In some environments, particularly those in which the same types of queries are frequently used, SEED's greater efficiency in retrieval will be an advantage. Literature from the vendor indicates that a user interface, which will permit the user to specify data in relational forms, has been made available. Since the difficulty in writing SEED schema definitions was one of the major problems in testing and evaluating the system, use of the relational user interface could make SEED a more effective DBMS.

Like the other two systems, SEED provides a full range of update/load facilities, including the ability to modify and edit individual data items, and to add or delete records. A bulk load facility is provided. Relational and Boolean operators may be used to qualify record specifications for modification or deletion. Update and load functions may be performed from application programs.

SEED has a full range of query and report components, including the ability to perform query and report functions from a programming language. Unlike other network-oriented systems, SEED permits the use of relational and Boolean operators to qualify records or data items for retrieval.

Restart and recovery components are provided, including an audit log, the ability to dump or copy data base contents, and a checkpoint facility, with rollback and rollforward capabilities. Facilities for data security are provided.

Concurrency controls permit multiple users to access the same collection of data at a lower level than the entire data base, e.g. file, set, or relation. Data definition components include the ability to define schemas and subschemas. SEED also supports facilities for reorganization of the data base for greater storage and retrieval efficiency.

SEED includes a monitoring facility to determine and report on the status of the SEED files, such as the number of records, percent filled, etc. SEED will support large data bases (one billion bytes or larger) and multi-volume files. Memory requirements are approximately 50K words. SEED has the lowest cost among the three systems tested, estimated at $14,000 at the time of evaluation.

Complete documentation for SEED was provided, including a user manual, programming manual, and training materials. Training was provided on-site. The vendor has developed a particularly effective course of training, with a variety of approaches and many carefully-planned student exercises. However, many of the examples used in training were too simple to provide guidance for later use of the system, and the level of training provided seemed to be inconsistent.

SEED is implemented in Fortran, with IBM, Modcomp, CDC, TOPS-10, and VAX versions available. The version used for testing was modified by the vendor for the IAS operating system for the PDP-11/70. Some problems in system performance may be traced to the need for using this modified system, rather than one which was specifically developed for IAS.

In the initial evaluation, SEED rated nearly as high as ADABAS-M and ORACLE, and higher than most of the other systems considered. It was rated highly for its low cost and portability. For applications in which these factors are important, SEED would have an even higher rating. It was regarded as less impressive in its update, load, query, and report functions, and in its security and concurrency facilities.

11.0 USE OF SYSTEM

This section describes the procedures that were followed in setting up and running the Benchmark Driver. In addition to outlining the design of the RTE approach, it will provide guidance for future users of the system.

In order to execute a test run, a series of steps in precise order must be performed. These steps ensure that the system is configured correctly and that all of the tools to capture the resulting statistics for evaluation are operative. A checklist was developed and followed to ensure that the steps were accomplished in their proper order.

Step 1: Position the data base to its ground zero position: At the beginning of each test run, the contents of the data base must be restored to the same initial values. This will ensure that any variations in results are due to the load imposed upon the DBMS from the scenarios and not to any difference in data base contents. ADABAS-M had to be shut down before any of the files could be restored. Indirect command files were constructed containing the series of commands to restore the proper files, named RESTOREX.CMD in UFD [7,7]. To execute these files, "RESTOREX" is typed, where X is the number of the scenario that is being tested. All files used by that scenario were restored. Similar command files were constructed for ORACLE to simplify the restoration process.

Step 2: Bring up the DBMS: Since ADABAS-M had to be shut down before any of the files could be restored to their baseline position, ADABAS-M had to be activated and all files used by the scenarios under test had to be opened before any test could be initiated. In ORACLE, it was not necessary to shut down the system in order to restore the data base; hence this step does not apply to ORACLE runs.

Step 3: Restoration of the Seed files: The Seed file contained the seeds used for the random number generator called by the scenarios. The Benchmark Driver would access the seed file and select the next unused seed in the file each time a random number generator needed to be seeded during execution of the scenarios. As each seed was selected from the file, it was marked by the BMD so that the same seed was not reused in that specific test run. These flags, marking the used seeds, had to be cleared before each test run, and the order of the seeds had to be set to ensure that each test run could be duplicated exactly in each DBMS under test. A short program was written to allow the user to set the seed file with the sequence of seeds to be used.

Step 4: Set up the General Purpose Monitor input parameters: Files containing the input parameters for all of the different options in the GPM for the different scenarios were built and stored. For the specific scenarios under test, the appropriate files were recalled and used as input for the GPM.

Step 5: Install the proper number of tasks for each scenario for ADABAS-M: The ADABAS-M IT pseudo-device handler cannot handle multiple occurrences of the same task name. In RSX-11, under which the IT driver was developed, duplicate names are not allowed. This is not so in IAS. Duplicate task names may exist; tasks are differentiated by their Job Number. If multiple copies of the Collection Coordination scenario with the same name exist, the packet of information will go to the first Collection Coordination scenario task, which may not be the correct copy. For the RTE, the BMD was designed and coded for multiple copies of each scenario. Therefore, each task or copy of a scenario will have to be installed under a unique name, limiting the number of copies which may exist at any one time.

Step 6: Set up the Benchmark Driver: The Benchmark Driver (BMD) is the program that will begin a testbed session. The user will run this program to set scenario and script parameters, then execute a session. To execute the Benchmark Driver, BMD is entered at the terminal. The BMD program will respond with the prompt:

```
BMD>
```

requesting the user to enter the desired parameter. The possible commmands or parameters are:

SC	Set control file parameters
SA	Save the display
ED	Edit the current display
PR	Print the display on the line printer
RU	Execute the Benchmark test run.

The sequence of commands to execute a typical evaluation run is as follows:

a) Execute the SC command to review the parameters saved in the file from a previous test run. By the saving the parameter files, the run could be duplicated exactly at a later time if desired. If modifications of the parameters in the stored file were desired the user would proceed to Step b. Otherwise, the user would proceed to Step e.

b) Execute the ED command to edit the current parameters.

c) Execute the SA command to save the current parameters.

d) Execute the PR command to print the parameters at the line printer. When the output of the GPM was printed at the completion of the test run, a complete hardcopy record was thus produced.

e) The Run command is not invoked until the GPM has been initiated.

Step 7: Start the GPM.

Step 8: Start the actual execution of the scenarios by executing the Run command of the BMD.

Step 9: Stop the GPM: At the completion of the test, stop the GPM, format the raw statistics file for printing, print the statistics for the test run on the line printer.

12.0 BENCHMARK TEST METHODOLOGY

RTE testing was first introduced during the late 1960s in the form of a sophisticated system driver to simulate multiple interactive users in a time-shared computer system. The RTE software emulates (i.e. attempts to reproduce) the behavior of terminal users as they send commands to the System Under Test (SUT) and receive responses from the SUT.

By emulating the behavior of the terminal users, the RTE approach permits an inexpensive, controlled, repeatable series of system tests. It contrasts with live testing, which is expensive (in terms of personnel and equipment times), uncontrolled (human actions cannot be completely controlled), and unrepeatable (human behavior is not precisely repeatable).

An RTE is capable of loading the system with a large number of simulated user terminals, which operate at a controlled rate, to permit testing of relevant system characteristics, such as response times, resource requirements, and behavior under overload. The RTE tests the ability of the DBMS to respond to the needs of I&W analysts accessing the data base through their terminals.

Some typical analyst tasks may be listed as follows:

- Review messages
- Consult with other sources
- Issue an alert
- Flag an item for further investigation
- Route messages

- Access data bases

- Correlate data about this situation

- Analyze this situation, i.e. determine its meaning, relevance, or implications

- Project near-term estimates of intensity, scope, and direction of threat

The purpose of the RTE benchmark tests is to provide statistical information concerning operation of each of the systems in the performance of these tasks to permit the identification of significant differences among the systems.

Proper procedures for a benchmark test require that testing conditions for each of the systems will be the same. Several factors made this experimental design unworkable. Specifically, errors in software supplied for all three systems, particularly those errors that led to the withdrawal of ADABAS-M from the market, together with the intractability of the SEED schema definitions, seriously curtailed the benchmark evaluation process. Although the evaluation might have stopped at this point, it was felt that a portion of the RTE should be completed. In this way, the test facility could be demonstrated and valuable information could be obtained concerning DBMS characteristics. The only major limitation would be that a full statistical analysis of differences among the systems would have to be omitted.

The RTE Benchmark Driver initiates scripts, submitting them to the DBMS under test. While executing, the scripts provide output resembling that of an analyst at a terminal, which is submitted to the system under test. Operation of the BMD is monitored by the General Purpose Monitor, which provides statistical information concerning system performance.

The Benchmark Driver is under the control of the user, who submits commands to it for execution. The following paragraphs briefly describe each of the user commands. In all cases, brackets [] enclose optional fields and parentheses () enclose alternate names.

SET-CONTROLS(SC)=[filename]

This command retrieves a set of control parameters using the given or the default filename together with suffix ".CP" (if no suffix is supplied, ".CP" is assumed). The control parameters from this file are displayed. The default filename is set to the given filename, if present. Subsequent file accesses will be prefixed with this default filename until another SC command is executed. (EXAMPLE: SC = EXER00).

S-ITEMS (SI) = [filename.][suffix]

This command will cause the retrieval and display of the S-Item parameter set contained in the file with a name equal to the given (or default) filename together with the given suffix. The suffix to be used to represent one of the six scripts must be S1, S2, S3, S4, S5, or S6. For example, S-Items previously SAVEd for script one could be recalled using the default filename by using the command: S-ITEMS=S1. If more than one parameter set has been saved under this filename, the system returns the first set found.

SAVE=[filename.][suffix]

The SAVE command stores the values from the currently displayed menu into a file with a name equal to the given (or default) filename together with the given suffix or the default suffix (for the data currently being displayed). (EXAMPLE: SAVE=EXER01). To avoid later confusion, care should be taken that filename:suffix pairs are unique.

PRINT=[filename,][suffix]

This command results in a hardcopy being printed of the contents of the given (or default) filename using the given or default suffix (for data currently being displayed). (EXAMPLE: PRINT=S2FILE,S2).

STATS=filename

This command causes the hardcopy production of formatted statistical reports us;ng the statistics data stored in "filename". (EXAMPLE: STATS=STAT01).

RUN=[filename]

This command causes the start of a Benchmark Exercise by using the given (or default) filename together with suffix ".CP" to access the file of control parameters to use for the exercise. Based upon the setting of the control parameter FLAG fields, the "S-Item-FILE" filename will be used with the appropriate suffix for each of the "active" scripts to access the set of S-Items for each script. (EXAMPLE: RUN=EXER01).

HALT

This command causes the current Benchmark Exercise to cease execution temporarily.

RESUME

This command allows a previously HALTed Benchmark Exercise to be restarted.

STOP

This command causes the Benchmark Exercise to be terminated and statistics to be computed and stored.

13.0 BENCHMARK TEST RESULTS AND ANALYSIS

This section presents detailed results of the Benchmark Tests performed during this project. In general, execution of benchmark tests was monitored by the General Purpose Monitor to provide comparative statistics concerning DBMS performance.

Several developments have affected the format in which results are presented:

- As noted elsewhere in this report, problems with the implementation of ADABAS-M have caused the vendor to withdraw the system from further sale.

- ORACLE 2.3, used for the tests, has been superseded by ORACLE 3.0, which is said to have greatly improved performance characteristics.

- Problems in preparing schema definitions for SEED curtailed the full development of a data base for this system. As a result, only a very limited number of tests of this DBMS were possible.

The impact of these restrictions on the test design has been to place greater emphasis on human factors and other characteristics which cannot be directly measured, such as quality of documentation. An evaluation of these factors is needed to meet the complaints and requirements originally stated by I&W analysts. A major contribution of the evaluation project may well be the discovery of the severe limitations of the hardware and software that are available for support of I&W analysis.

With these caveats, the results of the benchmark tests are presented in this section.

13.1 Benchmark Execution and Results

Six of the scenarios described in Section 3.0, representing various types of analyst activities, based upon the following six intelligence tasks, were defined in DHIL and translated into scripts:

- Watch
- I&W Analysis
- Collection Coordination
- Area Analysis
- Military Systems Analysis
- MAC Route Assessment

To simulate variations in the performance of the tasks, pseudo-random variations are introduced into the scripts. The same set of pseudo-random variations is used for each of the systems under test, to permit direct comparisons among the systems. In addition:

- Varied levels of alert (peace, crisis, war) are simulated.
- Varied loads are introduced, including variations in the number of terminals and in the number of scripts processed at each terminal.

A universal top-level software translator was designed to translate the DHIL scenarios into Fortran code with embedded DBMS-specific calls. The design of the translator was not implemented in code because of the complexity of the task, given the requirement to be absolutely fair to each of the DBMSs under evaluation. Instead, scenarios were hand-translated into executable code. A given set of procedures, developed during the translator design process, were followed by the implementor for each scenario translation. These procedures were the same for each DBMS. Only details of implementation were different.

For ADABAS-M the files which must be opened for each scenario are included in Table 13-1. ADABAS-M limits the number of files that can be open at the same time. As the Watch scenario was originally defined, it required 18 files to be opened. Some initial tests of the RTE exceeded the upper limit for ADABAS-M, when an attempt was made to run the Collection Coordination and Watch scenarios concurrently. Because of the importance of running multiple scenarios, some calls to the data base were eliminated from each scenario. Version 2 of the Watch scenario uses only 14 files, to allow room for storage of files required by cther scenarios. Files A001B, A001C, A001D, and A001E were eliminated. Identical modifications were made for each DBMS.

In addition, files C001B and C001C were eliminated from the demonstration version of the system, to permit the WEIS file to be opened at the same time. In the demonstration, WEIS was used on an interactive basis, using cartographic displays, with the RTE operating in the background, to illustrate the effect of multiple tasks competing for system resources.

Three versions of the Watch Function scenario were fully implemented:

- o Version 1, used when no other scenarios were in use, operated with 18 files.

- o Version 2, used when other scenarios required data storage areas, operated with 14 files.

- o Version 3, used during demonstrations of the system, operated with 13 files, to permit use of the WEIS data.

TIB Subject	File Name
Messages	A001A, A001B, A001C, A001D, A001E, A1TEXT
Events	B001A, B001B
Order of Battle	C001A, C001B, C001C, C001D
Personalities	D002F
Country Summary	E001A
Indicator Lists	F001A, F001B
Friendly Mission Data	H001A
Air Route Segments	I002A

Table 13-1 Files Used by Watch Function Scenario

The files required for operation of the system for Version 1 are shown in Table 13-1. The Task Information Blocks (TIBs) are described in more detail in Appendix D.

The Collection Coordination scenario was the first to be completely implemented, because it required the smallest number of files. Two versions were implemented under ADABAS-M. Version 1 would update the G-file (Collection Request Form) after processing, while Version 2 would update and delete the collection request after processing the request. Both versions were executing under ADABAS-M and could run in conjunction with the Watch scenario. These modifications to the scenarios were applied to tests for all three DBMSs, thus presenting equal loads and variations of activity. Section 13.2 presents suggested enhancements to the DBMSs which, if implemented, would make the modifications to the scenarios unnecessary.

A special Update scenario was developed to add records or requests to the G-file. This scenario simulates the process by which analysts add collection requests for the Collection Coordination analyst to process. Each time this scenario was executed, it would add 50 collection requests to the file.

All modules of the Area Analysis scenario were designed, coded and tested successfully, using a small subset of the files developed specifically for testing. Because of the number and size of the modules, this scenario could not be successfully taskbuilt, since the size of the object code exceeded the maximum size allowed per program. A solution to this problem was to break up the modules even further, and to implement an overlay structure. The initial design, as implemented in code, would require recoding and retesting, taking into account the required overlay structure. The modules, as delivered, have all been tested successsfully and are executing correctly, but the overlay structure has not been implemented at this time, and the scenario has not been used in the RTE.

The modules included in the I&W Analysis scenario have all been coded and partially tested. Due to the errors in ADABAS-M and its loader, the 16 files required to complete, test, and execute this scenario were not all loaded within the time limits of this project, and no further testing of this scenario took place.

Individual modules of the MAC Route Assessment and Military Systems Analysis scenarios were coded. Difficulties with the ADABAS-M loader led to suspension of further tests of these scenarios.

Table 13-2 shows the implementation status of each of the scenarios for ADABAS-M.

Translation of the scenarios into FORTRAN programs for use with SEED was not completed because of the difficulty and time required to define schemas for this purpose. The following steps were carried out in order to define SEED schemas and to select a scenario for implementation. The general approach to schema definition for SEED consisted of two parts:

o A first-level design using a basic schema design diagram. The vendor evaluated the first-level design of a representative TIB, with comments and recommendations to be incorporated into implementations of the remaining TIBs.

o Entering and compiling schemas, and making adjustments to meet system limitations.

The complexity of a SEED schema is a function of the number of keys, the number of repeating groups, and the number of items found in the file.

Scenario Name	Status
Watch	Version 1 (18 files): implemented Version 2 (14 files): implemented Version 3 (demo, 13 files): implemented
Collection Coord.	Version 1: implemented Version 2: implemented
Area Analysis	Modules tested individually and running Overlays necessary for implementation
MAC Route Assessment	Coded but not completely tested
I&W Analysis	Coded but not completely tested
Military Systems	Coded but not completely tested
Update Message File	Coded
Modified Update	Implemented

Table 13-2 Implementation Status of Scenarios for ADABAS-M

Table 13-3 indicates the complexity level of TIB implementation for the scenarios. In addition, the number of items is indicated. If the number of items is less than 50, it is coded small (S); between 50 and 99, medium (M); between 100 and 200, large (L); and greater than 200, very large (VL).

A first-level design layout of TIB E002, Installation Information, which did not take into account any system limitations, was completed after consultations with the vendor. E002 was selected for the following reasons:

- It is one of the largest and most complex of the TIBs.
- It is indicative of all of the the problems or features to be found in the rest of the TIBs.
- It has a large number of keys and repeating groups.

The initial design was forwarded to the vendor for evaluation and comment. No faults were found in the design.

In order to perform a fair, unbiased evaluation of the three DBMSs, the same number of total hours, 160, was allocated to the schema definition task for each DBMS under evaluation. Since it was apparent that SEED would require considerably more than this allocation, the complexity estimates were used to assist in determining how best to distribute resources for the schema definition task. Table 13-4 indicates the TIBs used by each scenario. The complexity factor is enclosed in parentheses.

Since the Collection Coordination scenario uses the smallest number of TIBs, it was completed first for all three systems. In this way preliminary performance statistics used for system checkout and for comparisons were obtained before all of the scenarios were translated and

TIB	NUMBER OF KEYS	NUMBER OF REPEATING GROUPS	NUMBER OF ITEMS	COMPLEXITY LEVEL	
J001	2	–	S	1	EASIEST
F001	5	1	S	1	
I001	3	2	S	2	
A001	1	4	S	2	
H001	21	–	S	3	
E001	15	–	S	3	
D001	12	4	VL	4	
I002	25	3	L	5	
G001	26	2	L	5	
B001	28	8	M	5	
C001	35	8	M	5	
F002	28	26	L	6	(To SEED for evaluation)
D002	92	12	L	7	Most Difficult

Table 13-3 TJB Schema Complexity Level

TIB	WATCH FUNCTION	AREA ANAL	I&W ANAL	MAC ROUTE ASSESSMENT	COLLECTION COORDINATION	MIL-SYS. ANAL.
A001	XXX(2)	XXX(2)	XXX(2)	.	.	XXX(2)
B001	XXX(5)	.	XXX(5)	XXX(5)	.	.
C001	XXX(5)	XXX(5)	XXX(5)	XXX(5)	.	.
D001	.	XXX(4)	XXX(4)	XXX(4)	.	XXX(4)
D002	XXX(7)	.	.	.	.	.
E001	XXX(3)	.	.	XXX(3)	.	.
F001	XXX(1)	XXX(1)	XXX(1)	XXX(1)	.	.
G001	.	XXX(3)	.	.	XXX(5)	XXX(5)
H001	XXX(3)	XXX(3)	XXX(3)	XXX(3)	XXX(3)	XXX(3)
I001	XXX(2)	.	.	XXX(2)	.	.
I002	XXX(5)	XXX(5)	XXX(5)	XXX(5)	.	XXX(5)
J001	.	.	.	.	XXX(1)	.
TOTAL TIBS USED	9	7	7	7	3	6

Note: "XXX" indicates that the TIB is used for the given scenario. Complexity estimates are given in parenthesis.

Table 13-4 TIBs Per Scenario

all TIBs loaded. SEED schema definitions proved to require a great deal of time, and the choice of other scenarios for SEED tests was critical. While the Military Systems Analysis scenario uses only six TIBs, one of these is TIB E002, one of the most difficult TIBs to implement. With the allocation of time for SEED schema definitions more than exhausted, it was necessary to proceed with an evaluation based on a limited set of input queries.

The following evaluation criteria, extracted from the SOW (4.3.2) were applicable to tests of SEED, using a limited set of queries:

- o Data Base Creation
- o Data Base Update
- o Storage and Retrieval of Data
- o Response to Analyst Queries
- o Ease of Use
- o Overall Performance

It was apparent that these key DBMS evaluation criteria could be observed and measured for SEED. DBMS performance could be monitored by the General Purpose Monitor (GPM) to obtain the technical factors, such as disk activity, accesses per query, and CPU overhead. Behavior of the system under saturation and overload conditions could be determined by using a number of replications of the same scenario, rather than a mixture of all scenarios, as orginally planned. Such factors as the ease with which commands can be written, their readability, and many specific problems encountered in the use of the language are independent of the number of scenarios translated. Quantitative measures, such as labor hours required for design and machine time required to load the data base could be made.

Difficulties in writing schema definitions in the SEED language or problems encountered in loading could be noted and documented. For these reasons, an evaluation of SEED could still be completed, even though only one scenario was translated and implemented.

Errors in SEED appear to be due to the need to use the system under the IAS operating system, which required software modifications performed by the vendor.

For ORACLE, all of the modules for each scenario have been coded. All were successfully tested individually before integration. Although programs for ORACLE were simpler to write, they tended to be much longer than those for ADABAS and SEED. The voluminous code associated with ORACLE DBMS calls forced most of the scenarios into an overlay structure. Table 13-5 shows the implementation status of each of the scenarios for ORACLE.

Table 13-6 describes each Test Series executed during the RTE Benchmark. The top half of the table for each Test Series shows the scenario(s) that were performed. The bottom half shows the activity generated against files of the data base.

Table 13-7 is organized by selected measurements indicating relative DBMS performance for each measurement.

A full presentation of all results, by test, is contained in Table 13-8.

Scenario Name	ORACLE Implmentation Status
Watch	Coded; overlays required
Collection Coord.	Two versions implemented
Area Analysis	Coded; overlays required
MAC Route Assess.	Coded; overlays required
I&W Analysis	Coded; overlays required
Military Sys. Anal.	Coded; overlays required
Modified Update	Implemented

Table 13-5 Scenarios for ORACLE

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: A

Scenario	No. Terminals
Collection Coordination Update Only (Seed 10)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	2	30			0				
H-File	1	1							
J-File	1	30							

Table 13-6 Test Series Summary Descriptions (Page 1 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: B

Scenario	No. Terminals
Collection Coordination Update Only (Seed 20)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	2	7			1	3			
H-File	1	0							
J-File	1	28							

Table 13-6 Test Series Summary Descriptions (Page 2 of 19)

Test ID: C

Scenario	No. Terminals
Add 50 Records to G-File (Seed 10)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File			50	50					

Table 13-6 Test Series Summary Descriptions (Page 3 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: D

Scenario	No. Terminals
Collection Coordination Update and Delete (Seed 10)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	2	30			0		0		
H-File	1	1							
J-File	1	30							

Table 13-6 Test Series Summary Descriptions (Page 4 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: E

Scenario	No. Terminals
Collection Coordination Update and Delete (Seed 20)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	2	7			1	3	1	3	
H-File	1	0							
J-File	1	28							

Table 13-6 Test Series Summary Descriptions (Page 5 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: F

Scenario	No. Terminals
Collection Coordination Update Only (Seed 40)	3

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	6	21			3	4			
H-File	3	2							
J-File	3	102							

Table 13-6 Test Series Summary Descriptions (Page 6 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: G

Scenario	No. Terminals
Collection Coordination Update Only (Seed 30)	1
Collection Coordination Update and Delete (Seed 10)	1
Add to G-File (Seed 10)	1
	3

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	4	72	50	50	1	1	1	1	
H-File	2	1							
J-File	2	63							

Table 13-6 Test Series Summary Descriptions (Page 7 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: H

Scenario	No. Terminals
Collection Coordination Update Only (Seed 40)	1
Collection Coordination Update and Delete (Seed 10)	1
Add to G-File (Seed 10)	1
	3

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	4	44	50	50	2	2	1	1	
H-File	2	1							
J-File	2	67							

Table 13-6 Test Series Summary Descriptions (Page 8 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: I

Scenario	No. Terminals
Collection Coordination Update Only (Seed 10)	5

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	10	81			3	6			
H-File	5	2							
J-File	5	156							

Table 13-6 Test Series Summary Descriptions (Page 9 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: J

Scenario	No. Terminals
Collection Coordination Update Only (Seed 30)	3
Collection Coordination Update and Delete (Seed 10)	2
Add to G-File (Seed 10)	1
	6

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	10	93	50	50	4	13	2	7	
H-File	5	156							
J-File	5	2							

able 13-6 Test Series Summary Descriptions (Page 10 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: K

Scenario	No. Terminals
Collection Coordination Update Only (Seed 50)	3
Collection Coordination Update and Delete (Seed 10)	2
Add to G-File	1
	6

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	10	66	50	50	5	15	2	7	
H-File	5	1							
J-File	5	151							

Table 13-6 Test Series Summary Descriptions (Page 11 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: L

Scenario	No. Terminals
Collection Coordination Update Only (Seed 50)	5
Collection Coordination Update and Delete (Seed 10)	2
Add to G-File	1
	8

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
G-File	14	≥66	50	50	≥5	≥15	2	7	
H-File	7	≥ 1							
J-File	7	≥151							

Table 13-6 Test Series Summary Descriptions (Page 12 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: M

Scenario	No. Terminals
Watch with C-File (Seed 10)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
A001A	1	1	0-1	0-1					
A1TXT	1	1	0-1	0-1					
B001A	0-1	0-420							
B001B	0-1	0-420							
C001A	0-3	0-3							
C001B	0-3	0-3							
C001C	0-3	0-3							
C001D	0-3	0-3							
D002F	0-1	0-1							
E001A	1-2	1-2			0-1	0-1			
F001A	0-1	0-400							
F001B	0-1	0-400							
H001A	0-1	0-11							
I002A	0-1	0-3							

Table 13-6 Test Series Summary Descriptions (Page 13 of 13)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: N

Scenario	No. Terminals
Watch with C-File (Seed 20)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
A001A	1	1	0-1	0-1					
A1TXT	1	1	0-1	0-1					
B001A	0-1	0-420							
B001B	0-1	0-420							
C001A	0-3	0-3							
C001B	0-3	0-3							
C001C	0-3	0-3							
C001D	0-3	0-3							
D002F	0-1	0-1							
E001A	1-2	1-2			0-1	0-1			
F001A	0-1	0-400							
F001B	0-1	0-400							
H001A	0-1	0-11							
I002A	0-1	0-3							

Table 13-6 Test Series Summary Descriptions (Page 14 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: 0

Scenario	No. Terminals
Watch without C-File (Seed 10)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
A001A	1	1	0-1	0-1					
A1TXT	1	1	0-1	0-1					
B001A	0-1	0-420							
B001B	0-1	0-420							
C001A	0-3	0-3							
C001D	0-3	0-3							
D002F	0-1	0-1							
E001A	1-2	1-2			0-1	0-1			
F001A	0-1	0-400							
F001B	0-1	0-400							
H001A	0-1	0-11							
I002A	0-1	0-3							

Table 13-6 Test Series Summary Description (Page 15 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: P

Scenario	No. Terminals
Watch without C-File (Seed 20)	1

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
A001A	1	1	0-1	0-1					
A1TXT	1	1	0-1	0-1					
B001A	0-1	0-420							
B001B	0-1	0-420							
C001A	0-3	0-3							
C001D	0-3	0-3							
D002F	0-1	0-1							
E001A	1-2	1-2			0-1	0-1			
F001A	0-1	0-400							
F001B	0-1	0-400							
H001A	0-1	0-11							
I002A	0-1	0-3							

Table 13-6 Test Series Summary Descriptions (Page 16 of 19)

Test ID: Q

Scenario	No. Terminals
Simple Query (Retrieve on one key)	0

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
J-File	100	3417							

Table 13-6 Test Series Summary Descriptions (Page 17 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: R

Scenario	No. Terminals
ANDED Query (Retrieve on two keys ANDed)	0

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded
J-File	100	19							

Table 13-6 Test Series Summary Descriptions (Page 18 of 19)

I&W DBMS ANALYSIS
TEST DESCRIPTION

Test ID: Z1 through Z5

Scenario	No. Terminals
LOAD	0

Files Affected	Searches	No. Recs	Adds	No. Recs	Modifies	No. Recs	Deletes	No. Recs	No. Recs Loaded	
J-File									4950	#1
J-File									2475	#2
J-File									1238	#3
J-File									619	#4
J-File									310	#5

Table 13-6 Test Series Summary Descriptions (Page 19 of 19)

I&W DBMS ANALYSIS

RELATIVE COMPARISONS

ELAPSED TIME (SECONDS)

DBMS TEST SERIES	ADABAS-M	ORACLE	SEED
A	147	310	--
B	218	326	--
C	130	794	--
D	147	311	--
E	226	359	--
F	272	427	--
G	322	818	--
H	315	1,083	--
I	307	582	--
J	567	1,537	--
K	426	1,420	--
L	ABORTED	3,707	--
M	1,780	--	--
N	544	--	--
O	1,764	--	--
P	543	--	--
Q	1,153	830	193
R	54	414	396/366
Z1	316	7,726	23,181
Z2	198	3,811	9,747
Z3	132	1,877	4,181
Z4	114	928	2,016
Z5	103	460	763

Table 13-7 Relative Comparisons (Page 1 of 6)

I&W DBMS ANALYSIS
RELATIVE COMPARISONS
CPU ACTIVITY (SECONDS)

DBMS TEST SERIES	ADABAS-M	ORACLE	SEED
A	36	89	--
B	32	94	--
C	86	532	--
D	36	88	--
E	50	114	--
F	75	178	--
G	157	687	--
H	150	688	--
I	91	318	--
J	250	962	--
K	252	929	--
L	ABORTED	1,424	--
M	868	--	--
N	160	--	--
O	881	--	--
P	159	--	--
Q	579	474	111
R	31	272	217/211
Z1	152	5,188	13,811
Z2	87	2,557	5,798
Z3	54	1,268	2,475
Z4	41	630	1,193
Z5	35	314	455

Table 13-7 Relative Comparisons (Page 2 of 6)

I&W DBMS ANALYSIS

RELATIVE COMPARISONS

FILE READ/WRITE SERVICE REQUESTS (RVB, RLB, WVB, WLB)

DBMS TEST SERIES	ADABAS-M	ORACLE	SEED
A	147	220	--
B	195	451	--
C	3,063	2,235	--
D	147	235	--
E	341	745	--
F	291	803	--
G	4,228	3,315	--
H	4,082	3,270	--
I	556	1,086	--
J	5,893	5,303	--
K	5,917	5,465	--
L	ABORTED	6,839	--
M	6,114	--	--
N	912	--	--
O	5,885	--	--
P	885	--	--
Q	5,991	3,754	2,037
R	318	2,678	4,083/3,922
Z1	4,240	46,512	160,130
Z2	2,269	21,637	56,763
Z3	1,349	9,879	21,497
Z4	926	4,408	10,592
Z5	726	1,874	2,752

Table 13-7 Relative Comparisons (Page 3 of 6)

I&W DBMS ANALYSIS

RELATIVE COMPARISONS

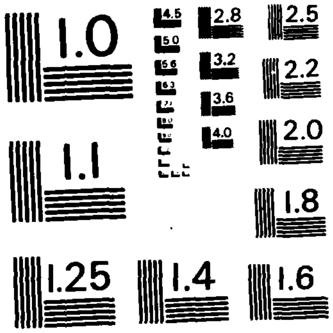
PHYSICAL DISK I/Os

DBMS TEST SERIES	ADABAS-M	ORACLE	SEED
A	--	--	--
B	697	840	--
C	--	4,131	--
D	--	--	--
E	--	--	--
F	2,790	--	--
G	12,391	--	--
H	11,691	--	--
I	3,501	--	--
J	19,570	--	--
K	--	10,669	--
L	ABORTED	--	--
M	--	--	--
N	--	--	--
O	--	--	--
P	--	--	--
Q	24,692	7,767	4,167
R	est. 1,081	3,368	8,259
Z1	7,287	89,446	436,029
Z2	4,296	41,394	164,787
Z3	2,904	18,801	65,907
Z4	2,307	8,327	30,555
Z5	2,020	3,450	8,881

Table 13-7 Relative Comparisons (Page 4 of 6)

AD-A139 339 I&W (INDICATIONS AND WARNING) DATA BASE MANAGEMENT 2/4
SYSTEMS ANALYSIS(U) MEASUREMENT CONCEPT CORP ROME NY
JUN 83 RADC-TR-83-136 F30602-80-C-0286

UNCLASSIFIED F/G 5/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

I&W DBMS ANALYSIS
RELATIVE COMPARISONS
LOAD OVERLAYS

DBMS TEST SERIES	ADABAS-M	ORACLE	SEED
A	177	31	--
B	177	25	--
C	156	15	--
D	177	33	--
E	159	29	--
F	474	54	--
G	2,574	50	--
H	2,256	58	--
I	1,164	70	--
J	5,354	74	--
K	4,406	60	--
L	ABORTED	78	--
M	6,230	--	--
N	780	--	--
O	6,204	--	--
P	780	--	--
Q	7,054	25	36
R	256	32	32/39
Z1	372	44	53,917
Z2	376	40	25,473
Z3	367	34	11,354
Z4	368	39	4,120
Z5	366	34	1,672

Table 13-7 Relative Comparisons (Page 5 of 6)

I&W DBMS ANALYSIS
RELATIVE COMPARISONS
COMPUTE VS. I/O*

DBMS TEST SERIES	ADABAS-M	ORACLE	SEED
A	--	--	--
B	1.15	2.80	--
C	--	2.68	--
D	--	--	--
E	--	--	--
F	.67	--	--
G	.32	--	--
H	.32	--	--
I	.65	--	--
J	.32	--	--
K	--	2.18	--
L	ABORTED	--	--
M	--	--	--
N	--	--	--
O	--	--	--
P	--	--	--
Q	.59	1.53	.67
R	--	2.02	.66
Z1	.52	1.45	.79
Z2	.51	1.54	.88
Z3	.46	1.69	.94
Z4	.44	1.89	.98
Z5	.43	2.28	1.28

*(CPU total secs.) divided by (# physical I/Os times 40ms)

Table 13-7 Relative Comparisons (Page 6 of 6)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: A

DBMS ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	147	310	
CPU - TOTAL* (secs)	36	89	
CPU - DBMS (secs)	5	25	
RVB	147	8	
RLB	0	175	
WVB	0	16	
WLB	0	21	
LOAD OVERLAY	177	31	
DISK IOs			

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 1 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: B

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	218	326	
CPU - TOTAL* (secs)	32	94	
CPU - DBMS (secs)	4	26	
RVB	128	8	
RLB	28	270	
WVB	30	25	
WLB	9	148	
LOAD OVERLAY	177	25	
DISK IOs	697	840	

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 2 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: C

DBMS ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	130	794	
CPU - TOTAL* (secs)	86	532	
CPU - DBMS (secs)	28	238	
RVB	1,901	1	
RLB	0	220	
WVB	1,162	513	
WLB	0	1,501	
LOAD OVERLAY	∿156	15	
DISK IOs		4,131	

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 3 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: D

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	147	311	
CPU - TOTAL* (secs)	36	88	
CPU - DBMS (secs)	5	26	
RVB	147	6	
RLB	0	184	
WVB	0	21	
WLB	0	24	
LOAD OVERLAY	177	33	
DISK IOs			

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 4 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: E

DBMS ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	226	359	
CPU - TOTAL* (secs)	50	114	
CPU - DBMS (secs)	5.4	34	
RVB	242	5	
RLB	21	458	
WVB	86	40	
WLB	2	242	
LOAD OVERLAY	159	29	
DISK IOs			

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 5 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: F

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	272	427	
CPU - TOTAL* (secs)	75	178	
CPU - DBMS (secs)	11	59	
RVB	251	12	
RLB	0	508	
WVB	40	30	
WLB	0	253	
LOAD OVERLAY	∿474	54	
DISK IOs	2,790		

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 6 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: G

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	322	818	
CPU - TOTAL* (secs)	157	687	
CPU - DBMS (secs)	44	304	
RVB	2,701	13	
RLB	0	1,174	
WVB	1,527	525	
WLB	0	1,603	
LOAD OVERLAY	∿2,574	50	
DISK IOs	12,391		

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 7 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: H

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	315	1,083	
CPU - TOTAL* (secs)	150	688	
CPU - DBMS (secs)	41	288	
RVB	2,574	16	
RLB	0	1,246	
WVB	1,508	531	
WLB	0	1,477	
LOAD OVERLAY	∿2,256	58	
DISK IOs	11,691		

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 8 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: I

DBMS ACTIVITY	ADABAS-M	ORACLE	ED
ELAPSED TIME (secs)	307	582	
CPU - TOTAL* (secs)	91	318	
CPU - DBMS (secs)	20	115	
RVB	505	20	
RLB	0	716	
WVB	51	36	
WLB	0	314	
LOAD OVERLAY	∿1,164	70	
DISK IOs	3,501		

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 9 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: J

ACTIVITY \ DBMS	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	567	1,537	
CPU - TOTAL* (secs)	250	962	
CPU - DBMS (secs)	63	391	
RVB	3,840	20	
RLB	0	2,588	
WVB	2,053	579	
WLB	0	2,116	
LOAD OVERLAY	∿5,345	74	
DISK IOs	19,570		

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 10 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: K

ACTIVITY \ DBMS	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	426	1,420	
CPU - TOTAL* (secs)	252	929	
CPU - DBMS (secs)	64	344	
RVB	3.888	19	
RLB	0	2,637	
WVB	2.029	588	
WLB	0	2,221	
LOAD OVERLAY	4,406	60	
DISK IOs		10,669	

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 11 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: L

DBMS ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)		3,707	
CPU - TOTAL* (secs)		1,424	
CPU - DBMS (secs)	WOULD NOT EXECUTE	476	
RVB		28	
RLB		3,563	
WVB		612	
WLB		2,636	
LOAD OVERLAY		78	
DISK IOs			

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 12 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: M

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	1,780		
CPU - TOTAL* (secs)	868		
CPU - DBMS (secs)	335		
RVB	6,100		
RLB	0		
WVB	14		
WLB	0		
LOAD OVERLAY	∿6,230		
DISK IOs			

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 13 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: N

DBMS ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	544		
CPU - TOTAL* (secs)	160		
CPU - DBMS (secs)	45		
RVB	911		
RLB	0		
WVB	1		
WLB	0		
LOAD OVERLAY	~780		
DISK IOs			

*Attributable directl tly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 14 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: 0

DBMS ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	1,764		
CPU - TOTAL* (secs)	881		
CPU - DBMS (secs)	355		
RVB	5,872		
RLB	0		
WVB	13		
WLB	0		
LOAD OVERLAY	∿6,204		
DISK IOs			

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 15 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: P

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	543		
CPU - TOTAL* (secs)	159		
CPU - DBMS (secs)	44		
RVB	884		
RLB	0		
WVB	1		
WLB	0		
LOAD OVERLAY	780		
DISK IOs			

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 16 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: Q

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	1,153	830	193
CPU - TOTAL* (secs)	579	474	111
CPU - DBMS (secs)	198	161	32
RVB	5,988	1	1,840
RLB	3	3,715	197
WVB	0	18	0
WLB	0	20	0
LOAD OVERLAY	7,054	25	36
DISK IOs	24,692	7,767	4,167

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 17 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: R

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED	
			via place set	via country code set
ELAPSED TIME (secs)	54	414	396	366
CPU - TOTAL* (secs)	31	272	217	211
CPU - DBMS (secs)	14	105	33	61
RVB	315	1	2,966	3,577
RLB	3	2,671	1,117	345
WVB	0	6	0	0
WLB	0	0	0	0
LOAD OVERLAY	256	32	32	39
DISK IOs		3,368	8,259	

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 18 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: S

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	43		567
CPU - TOTAL* (secs)	25		312
CPU - DBMS (secs)	8.6		64
RVB	84		3,847
RLB	11		1,190
WVB	190		19
WLB	7		1
LOAD OVERLAY	253		89
DISK IOs	864		10,327

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 19 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: Z1

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	316	7,726	23,181
CPU - TOTAL* (secs)	152	5,188	13,811
CPU - DBMS (secs)	75	2,380	3,374
RVB	2,135	1,867	71,661
RLB	484	18,000	58,190
WVB	1,289	6,670	24,486
WLB	332	19,975	5,793
LOAD OVERLAY	372	44	53,917
DISK IOs	7,287	89,446	436,029

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 20 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: Z2

DBMS ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	198	3,811	9,747
CPU - TOTAL* (secs)	87	2,557	5,798
CPU - DBMS (secs)	38	1,174	1,468
RVB	1,005	933	21,312
RLB	298	7,736	19,005
WVB	631	3,251	13,760
WLB	335	9,717	2,686
LOAD OVERLAY	376	40	25,473
DISK IOs	4,296	41,394	164,787

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 21 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: Z3

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	132	1,877	4,181
CPU - TOTAL* (secs)	54	1,268	2,475
CPU - DBMS (secs)	19	585	662
RVB	488	470	7,136
RLB	223	3,157	6,575
WVB	319	1,571	6,634
WLB	319	4,681	1,152
LOAD OVERLAY	367	34	11,354
DISK IOs	2,904	18,801	65,907

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 22 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: Z4

ACTIVITY \ DBMS	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	114	928	2,016
CPU - TOTAL* (secs)	41	630	1,193
CPU - DBMS (secs)	11	292	308
RVB	241	238	2,666
RLB	205	1,225	4,379
WVB	167	743	3,086
WLB	313	2,202	461
LOAD OVERLAY	368	39	4,120
DISK IOs	2,307	8,327	30,555

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 23 of 24)

I&W DBMS ANALYSIS
TEST SUMMARY

Test Series: Z5

DBMS / ACTIVITY	ADABAS-M	ORACLE	SEED
ELAPSED TIME (secs)	103	460	763
CPU - TOTAL* (secs)	35	314	455
CPU - DBMS (secs)	6	146	132
RVB	126	122	1,137
RLB	190	439	87
WVB	95	333	1,386
WLB	315	980	142
LOAD OVERLAY	366	34	1,672
DISK IOs	2,020	3,450	8,881

*Attributable directly or indirectly to the DBMS

Table 13-8 I&W DBMS Analysis Test Summary (Page 24 of 24)

13.2 Comparative Analysis by Type of Activity

This paragraph presents comparisons of the three DBMSs' efficiency while performing three general types of activity: LOAD, RETRIEVE, and UPDATE. Multi-user effects are then examined using statistics gathered during execution of the multi-user I&W scenarios. A ranking of performance is followed by discussions of the sensitivity of performance to operational variables, the significance of observed and projected performance to I&W requirements, and recommendations for improvement. Tables 13-6 through 13-9 contain statistics used during this analysis.

13.2.1 LOAD

(Test Series Z1-Z5)

13.2.1.1 Comparison

To acquire statistics on LOAD performance, the J-File was loaded five times with each DBMS. The number of records loaded was doubled on each load for the purpose of developing a set of curves representing performance as a function of number of records loaded. An unordered flat-file input was considered the most neutral of formats and was therefore employed in all loads. It was discovered that SEED is painfully sensitive to input structure and ordering. The input it was presented with is perhaps the worst it could experience. With a small amount of effort, a user site could pre-process data to better suit SEED, probably bringing it up to the load efficiency of ORACLE as observed.

The unequivocal winner of the LOAD competition, however, is ADABAS-M. Its performance in LOADing, when compared to that of the other two DBMSs, is dramatic; for example, ADABAS-M took 316 seconds of elapsed time to load 4,950 records, compared to 7,726 seconds for ORACLE and 23,181 seconds for SEED. In addition, the elapsed time (and other measures) for ADABAS-M, as a function of number of records loaded appears to be very close to linear.

13.2.1.2 Sensitivity to Operational Variables

A projected figure of approximately 8,000 seconds for ADABAS-M to load approximately 150,000 records gives support to the vendor's claim of 100,000 records loaded in 1 1/2 hours. ORACLE appears to exhibit a slight increase in elapsed time per record as the number of records increases (.025 seconds increase per record per doubling), whereas SEED appears to incur an additional second per record for each doubling. Using these numbers to project would give 266,909 seconds (74 hours) for ORACLE and 1,494,965 seconds (415 hours) for SEED to load approximately 150,000 records. The SEED number, it must be remembered, is an absolute worst possible case.

13.2.1.3 Significance to I&W Requirements

Since loading the data base is essentially a one-time chore, the efficiency of this process is not terribly important, assuming time-to-load is kept within reasonable bounds. The 74 hours projected for ORACLE to load 150,000 records, for example, is considered reasonable in that it represents approximately three days of non-stop loading, a not unusual experience when loading intelligence data bases. As mentioned before, it is anticipated that SEED, with favorable organization of input, could perform just as well.

13.2.1.4 Recommendations for Improvement

Obviously, ADABAS-M is in no need of improvement for loading of data. It is apparent that ORACLE is maintaining index entry order during load. If this is true, efficiency could probably be enhanced dramatically by waiting until the end of input has been reached to sequence and organize the indices.

13.2.2 RETRIEVE
(Test Series A,D,M,N,O,P,Q,R)

13.2.2.1 Comparison

Retrieval results indicate that the basic access algorithm of ADABAS-M is inferior to both ORACLE and SEED, but that ADABAS-M's query resolution logic is superior. In the Simple Query (Test Series Q) where a majority of the records pass qualification (3,415 out of 4,950), SEED wins handily since it can CALC directly to owner records of interest and retrieve data records using pointers found in the owner records. In this case, all records read are known to pass qualification before they are accessed, so no effort is wasted. ORACLE must incur overhead in searching to acquire record pointers, but this overhead is smaller than that of ADABAS-M.

These observations are based primarily upon RVB (Read Virtual Block) and RLB (Read Logical Block) counts for the three DBMSs for Test Series Q (5,991, 3,716 and 2,037, respectively). The three time-associated measures (CPU-DBMS, CPU-TOTAL and ELAPSED TIME) show an even greater disparity, and support the previous analysis, indicating that there are not only more disk operations associated with indexing than with hashing (no suprise), but that, since the CPU times are greater per RVB/RLB performed in the indexing models, it can be supposed that the logic supporting indices is more involved and costly than that supporting hashed access.

A mystery persists in Test Series Q and was also observed in Test Series M and O. During this simple retrieval of 3,415 records, ADABAS-M performed 7,040 Load Overlays (LOV) (14 more were performed outside of ADABAS-M). There is no need in the single program user environment of this test for rolling programs in and out. Indeed, in Test Series R, identical to Q except for added query qualifications, this behavior was not observed. It was thought that perhaps ADABAS-M was using the LOV as a more efficient means for reading data records or indices. Since all of the LOV activity is to the system disk rather than the data disk, the use of LOV to read qualifying data records must be discarded. Since Test Series R must have accessed indices at least as much as Test Series Q, and since R employed only 256 LOVs, this hypothesis was also rejected.

One hypothesis which has not been rejected is that the ADABAS-M design emphasizes and expects a multi-user environment, and optimizes for this environment to the detriment of the single user, "batch", type of operation. It may automatically save its buffer by rolling it out to disk every time a new record comes in. This action would be taken under the assumption that the current buffer contents belong to a user other than the user who owns the record coming in. In this case, it cannot be assumed that the user of the original buffer is finished with its contents. The buffer must be saved. A simple check should be possible to discover whether or not the user has changed. If the same user is requesting another record, it can be assumed that the original buffer contents are no longer needed and, therfore, do not need to be saved.

The only other unrejected hypothesis is that the LOV activity is associated with maintenance of the bit map employed for recording hits. If for some reason a bit map is being rolled in and out, via LOV, each time a record qualifies, the activity could be explained. This explanation would require, however, that for a compound query such as Test Series R, ANDing on results of multiple key searches is performed before bit maps are rolled out.

The picture for compound queries is markedly different. In contrast to the analysis above concerning basic access algorithms, it can be observed in Test Series R that the query resolution algorithm employed by ADABAS-M is far superior to those of both ORACLE and SEED. In going from Q, which retrieves 3,415 qualifying records to R, which retrieves 19 qualifying records, ADABAS-M elapsed time drops by 95%, ORACLE by 50%, and SEED time doubles. All other measures are similar. It is obvious from the results of Test Series R that ADABAS-M resolves the entire query via its indices before reading data records. ORACLE appears to be reading data records based on intermediate results derived from part of the query, and then qualifying further by examination of the record.

The schema definition employed for SEED was the cause of SEED's poor showing in Test Series R. As defined, the search keys involved (place name and country code) had no direct relation to each other. Had the schema associated place names as a set subordinate to the country code set, SEED would have performed much better, employing a hash to country code and a short chain walk (in this data base) to the place name. It was felt, however, in the schema design phase of the project that requests for messages by location would normally be by either country code or place name, not both, since the place names of places of interest in an I&W data base are generally unique. Where they are not unique, the additional retrievals for duplicative place names were felt to be less of a burden to the system than extensive chain walking in a large data base that would occur if, for instance, TYARATAM were to be sought in the chain of all places recorded as being in the USSR.

Test Series A and D were Collection Coordination scenarios each simulating one terminal, starting with identical seeds. The difference in the two was that A was to modify qualifying records and D was to delete them. In the event, neither scenario found the correct combination of qualifying records from the G, H and J files to trigger its modifying or deleting activity. A total of 61 records were retrieved on simple criteria from the 3 files. ADABAS-M outperformed ORACLE approximately 2 to 1 in these tests. SEED was not tested.

13.2.2.2 Sensitivity to Operational Variables

ADABAS-M appears to be poorly suited to list processing types of activity, where long lists of records are manipulated. It is ideally suited for applications where queries are complex and the number of records qualifying is low. The bulk of ADABAS-M effort is expended in accessing data records rather than in finding them. From test results, it would furthermore seem likely that a larger data base would not seriously impair the performance of ADABAS-M in its searching activity.

ORACLE took only a little over twice as long to read 3,415 records as it did to read 61 or 19 records. Unlike ADABAS-M, ORACLE increased time from Test Series A to Test Series R, even though the number of records retrieved was smaller for R than for A. This indicates that ORACLE is more sensitive to the complexity of queries than to the number of records retrieved.

ORACLE's practice of reading data records before they are completely qualified could get it into serious trouble in a very large data base. Depending on the structure and distribution of the data base and the nature of queries, ORACLE could easily find itself performing tens of thousands of data record accesses needlessly, thus slowing response to the analyst drastically and saturating system resources for other requests. This effect has been avoided in intelligence data handling systems facing the same problem by allowing only pre-defined and formatted queries. In doing this, however, the flexibility and power of SEQUEL and the data model supported by ORACLE, probably its best features, would be neutralized.

SEED would be relatively insensitive to the size of the data base if all keys were CALC keys. Some degradation would occur, however, as the size of the data base increases relative to the available disk storage, due to hash collisions. As queries become more varied, it becomes more likely that keys will not be CALCed, thus requiring chain walking or unnecessary retrieval of records which satisfy only part of the query (as in ORACLE).

At this point, SEED becomes very sensitive to the distribution of data. Thus, if long chains exist, SEED will perform poorly on queries. Likewise, if relationships between data have not been established at schema definition time, SEED must perform extra work to associate them.

13.2.2.3 Significance to I&W Operations

Since I&W analysis is currently characterized by queries of a fairly standard and pre-determined nature, it is tempting to speculate that an appropriate data base design using SEED would be the most efficient approach. Two considerations, however, work against this decision. The first is that I&W activity to date has to a large degree been shaped by the tools it has had available to it. If a more flexible and powerful query tool were available, queries might not be so constrained. The other consideration is the growing potential importance of artificial intelligence to the I&W process. If artificial intelligence capabilities are to be introduced to enhance the effectiveness of the I&W analyst, they cannot be constrained by the inherent inflexibilities of the CODASYL model demonstrated on this project by SEED.

ADABAS-M performed best of all in the tests executed, except for the Q Series, which is not considered typical for I&W operations (long lists are seldom retrieved). In addition, ADABAS-M is perceived as supporting a more flexible query environment than SEED in that keys may be designated for all access requirements without running into limitations which are built-in (e.g., only one CALC field per record) or resorting to structures which become cumbersome in large data base environments (e.g., chains).

ORACLE, while not performing as well as ADABAS-M, appears to be less sensitive to the type of query and the most flexible of the three systems tested. This augurs well for the I&W environment, especially that of the future.

13.2.2.4 Recommendations for Improvement

ADABAS-M's problems with Test Series Q and excessive load overlay action should be investigated. It is quite likely that a minor modification could solve the observed inefficiencies. If the hypothesis stated above, that the LOV activity is due to an assumption that the environment will be heavily multi-user, is true, it may not pay to modify ADABAS-M, since the I&W environment is a musti-user environment. A trade-off investigation examining just how heavily multi-user the environment is, compared to the added CPU cost of knowing when a buffer need not be rolled out, compared to the CPU and I/O savings when roll-outs are avoided, should be performed before modifying ADABAS-M.

ORACLE would benefit from additional logic which would enable it to resolve entire queries before accessing data records, as ADABAS-M does. This would be a considerable effort in the best of cases. It is likely that the basic structures of ORACLE may not support such a modification.

13.2.3 UPDATE
(Test Series B,C,E,F,G,H,I,J,K,L)

13.2.3.1 Comparison

ADABAS-M performed better than ORACLE on every update scenario executed. Table 13-9 shows update test statistics for ADABAS-M and ORACLE, for the measures: elapsed time, total CPU time (attributable to the DBMS), and file service requests. The column "O/A" is the result of dividing the ORACLE statistic by the ADABAS-M statistic and shows the advantage (ratio larger than 1.0) or disadvantage (ratio smaller than 1.0) of ADABAS-M relative to ORACLE.

I&W DBMS ANALYSIS

TEST	ELAPSED TIME			CPU-TOTAL			FILE SERVICES			# COMMANDS	# RECORDS AFFECTED			
SERIES	A	O	O/A	A	O	O/A	A	O	O/A	ISSUED	RETR.	MOD.	DEL.	ADD
B	218	326	1.5	32	94	2.9	195	451	2.3	5	35	3		
C	130	794	6.1	86	532	6.2	3063	2235	.7	50				50
E	226	359	1.6	50	114	2.3	341	745	2.2	6	35	3	3	
F	272	427	1.6	75	178	2.4	291	803	2.8	15	125	4		
G	322	818	2.5	157	687	4.4	4228	3315	.8	60	136	1	1	50
H	315	1083	3.4	150	688	4.6	4082	3270	.8	61	112	2	1	50
I	307	582	1.9	91	318	3.5	556	1086	2.0	23	239	6		
J	567	1537	2.7	250	962	3.9	5893	5303	.9	76	251	13	7	50
K	426	1420	3.3	252	927	3.7	5917	5465	.9	77	118	15	7	50

Table 13-9 UPDATE Performance (ADABAS-M and ORACLE)

One overriding effect should be discussed. When ADDing records to the G-File, two separate relations had to be updated during the ORACLE tests due to the limitation on size of the SEQUEL working area. This caused extra command parsing and logic execution for ORACLE, handicapping it significantly.

The degree of this handicap can be seen in both elapsed time and total CPU time:

- The elapsed time advantage range (ADABAS-M to ORACLE) for tests having no ADDs (Test Series B,E,F and I) is 1.5 to 1.9. The range for total CPU time is 2.3 to 3.5.

- The elapsed time advantage range for tests which include the 50 record ADD (Test Series C,G,H,J, and K) is 2.5 to 6.1. The range for total CPU time is 3.7 to 6.2.

Additionally, it can be noted that there is a general trend for the advantage to drop as ADDs become a smaller proportion of total activity. Test Series C, which shows the greatest advantage for ADABAS-M (6.1, 6.2) performs only ADDs.

The only advantage shown for ORACLE is for file service requests during the ADD related tests, this advantage increasing as ADDs become a greater proportion of the activity. This indicates that although ORACLE has a relatively difficult time deciding what to do and how to do it, it is more efficient in actually storing new records. This condition of being relatively CPU bound is born out by page 6 of Table 13-7, which depicts CPU time versus physical I/O time. There is no suprise here. Since ORACLE supports a more general data model and more powerful (non-procedural) interface language, it can be expected to be more CPU intensive.

13.2.3.2 Sensitivity to Operational Variables

Examination of Table 13-10, which orders tests according to resouce utlization and workload, will show a very strong correlation of all three measures of ORACLE resource utilization to the number of commands issued to the DBMS. This supports comments made above concerning the expense involved in parsing a non-procederal language and supporting a very general and basic data model. The indication is one of almost direct correlation. ADABAS-M also shows a strong correlation of CPU time to the number of commands issued. Other ADABAS-M correlations are looser.

Of particular interest is the number of seconds of CPU time used by the two DBMSs per command. ADABAS-M takes from 1.7 to 4.0 seconds per command for more active jobs (C,G,H,I,J,K). It is in the 5.0 to 8.3 second range for less active jobs (B,E,F). ORACLE, for more active jobs, takes approximately 10.6 to 13.8 seconds per command. For less active jobs, it takes from 11.9 to 19.0 seconds per command. This means that ADABAS-M would saturate the CPU sometime before 35 requests per minute are made of it from resident drivers. It would saturate the CPU sometime before 12 requests are made of it from quick in and out drivers. ORACLE CPU best case saturation points are, respectively, 6 and 5 seconds.

Another observation should be made concerning the multi-user characterisics of ADABAS-M as they apply to update activity. It is the practice of ADABAS-M to maintain updated and new records in primary memory until they are forced out by space limitations or until ADABAS-M itself is terminated. This practice was observed during the benchmark tests when update tests were run to completion and their statistics blocks contained now writes to the data disk. The tests had to be re-run, with termination of ADABAS-M, to get valid statistics. ADABAS-M was still alive in memory, idling, waiting for other jobs to request its services. It was oblivious to the fact that its only user had terminated, and still maintained the

I&W DBMS ANALYSIS

	Tests in Ascending Order by Resource Utilization			Tests in Ascending Order by Workload		
	By Elapsed Time	By Total CPU Time	By # File Service Requests	By # Commands	By # Records	By # Users
ADABAS-M	C	B	B	B	B	B
	B	E	F	E	E	C
	E	F	E	F	C	E
	F	C	I	I	F	F
	I	I	C	C	H	G
	H	H	H	G	G	H
	G	G	G	H	K	I
	K	J	J	J	I	J
	J	K	K	K	J	K
ORACLE	B	B	B			
	E	E	E			
	F	F	F			
	I	I	I			
	C	C	C			
	G	G	H			
	H	H	G			
	K	K	J			
	J	J	K			

Table 13-10 Correlation of Resource Utilization to Workload

updated or added records in active buffers, assuming that until it absolutely had to put them on disk, it might as well keep them in primary memory in case another user requested them. Some savings may be had here if users tend to access the same records. This policy of maintaining updated and new records explains ADABAS-M's non-optional journaling of updates. If updates are not to be made permanent (written to disk) at the termination of the job which updated them, then some provision must be made to assure the user that the updates will eventually become permanent even in the eventuality of system failure sometime subsequent to successful termination of the update job. Automatic, non-optional logging of updates accomplishes this. This logging activity, of course, uses system resources, but it is certainly faster than immediate updates to the data base on disk. Logging activity can be observed, in fact, via the disk cylinder access and head movement histograms produced during the benchmark by the General Purpose Monitor, to be sequential, in-track writes, causing a minimum of costly head movement, usually none.

13.2.3.3 Significance to I&W Requirements

Since the I&W environment will likely continue to be that of a system of resident application drivers and *will include direct* calls to the DBMS from terminals, the CPU to command ratio should be in the low end. Thus, taking the best case for the more efficient DBMS, the system can be expected to support, at a theoretical maximum, 35 requests per minute. To maintain acceptable response times, this number should be lowered to approximately 25 requests per minute. This is probably in range for existing systems, but allows little room for an increase in the number of users and almost no support for advanced techniques of automatic inference and decision support. This is a generic problem to all data base management services provided by host resident software.

The automatic, non-optional journaling policy of ADABAS-M presents a question. Since it makes possible a savings when updated records are soon accessed again, it is potentially a resource saving policy. It can also be said that, since data integrity and security from loss are important considerations, that no resources are wasted; some form of journaling for recovery must in any case be provided. It is possible, however, that there may be a desire to avoid the cost of journaling on some files, as in the case of analyst work files, or possibly, to speed up processing during periods of urgent demand.

13.2.3.4 Recommendations for Improvement

It is felt that there is not much room for improvement in resource utilization in either ADABAS-M or in ORACLE. ADABAS-M appears rather efficient and ORACLE appears just as efficient considering the greater power and flexibility it affords. Unfortunately, this performance can probably not be enhanced sufficiently to meet algorithmic and level of support enhancements planned for I&W systems.

13.2.4 Multi-User Effects
(Test Series F,G,H,I,J,K)

13.2.4.1 Comparison

No discernable penalties were incurred either by ADABAS-M or ORACLE in the support of multiple users. Test Series F,G, and H all had three users. Both DBMSs showed marked increase in system resource utilization during G and H as compared to F, but G and H also performed four times as many commands as F. Test Series I, which has five users, performed more in the range of F than in the range of J and K, which had six users each. Again, the number of commands performed by I was closer to F than to G,H,J or K. Refer to Table 13-6 for information concerning the numbers and types of commands performed for each Test Series.

Although the number of users did not seem to have great effect, a case might be made that the number of tasks did have an effect in both ADABAS-M and ORACLE. The Test Series, which had by far the heaviest resource utilization (G,H,J,K) also had the highest number of tasks (3). It can be seen that I, which had more users than G and H, and also accessed more records than G and H, was much lower in resource utilization than they were. Test Series I also had, however, approximately one third of the number of commands to process. Again, the number of commands processed appears to be the determining factor.

As has been discussed earlier in this report, the benchmark tests were adversely affected by ADABAS-M's restriction of no more than 18 files open at one time. ORACLE's requirement for a separate copy of its user interface module for each terminal supported also impaired testing. It would seem that ADABAS-M is designed for a limited application set and ORACLE for a small number of users.

13.2.4.2 Sensitivity to Operational Variables

Although no clear correlation of efficiency to number of users was discerned, the functionality of the systems in supporting large numbers of users or large application sets is suspect. Although ADABAS-M might support a large number of users (this has not been proved), they must limit themselves to a total of 18 open files. ORACLE's requirement for embedding itself in user tasks could, at some point, cause a shortage of system memory.

13.2.4.3 Significance to I&W Requirements

ADABAS-M's 18 file restriction is severely detrimental to I&W requirements. Nct only does it put an unacceptable limit on expandability of I&W systems, it would not even now support system files and analyst work files at the same time.

ORACLE's memory eating propensity probably puts an unacceptable upper limit on the number of users.

13.2.4.4 Recommendations for Improvement

Probably the 18 file limitation in ADABAS-M is an arbitrary number which could be increased with a minimum of effort. The limitation is placed because a buffer is kept for each open file, thus using memory for each open file. If the subject I&W system has sufficient memory, there is no reason why an adequate number of files could not be supported. For example, assuming each buffer consists of 512 bytes, 100 open files could be supported at the cost of only 51,200 bytes of memory.

ORACLE's problem is more at the heart of its design philosophy (multi-tasking rather than multi-serving). It would probably be a major overhaul to change it.

14.0 QUALITATIVE EVALUATION FACTORS

This section contains qualitative evaluations of the ORACLE, ADABAS-M, and SEED systems from the point of view (1) of the user and (2) of the programmer. Other comments, dealing with specific system features, are included in the first subsection.

The material contained in this section is derived from a full year's experience with each of the systems under test, and reflects those features which may prove most significant in the actual choice of a system. Factors are included which appear to have the greatest potential impact on system selection. No single DBMS was found to be so outstanding that it can be recommended without qualification for all I&W analyst requirements. For this reason, this set of qualitative evaluations may be used to provide guidance in locating those features required by a particular installation.

Subsection 14.1 contains comments and descriptive material in narrative form. Subsections 14.2 and 14.3 contain evaluations in outline form. Recommendations from the point of view of user-analysts and programmers are contained at the end of these two subsections.

14.1 Qualitative Evaluations

This section contains narrative evaluations of the ORACLE, ADABAS-M, and SEED systems along the following dimensions:

- Query Language
- Interactive User Interface
- Host Language Interface
- Data Base Administrator Utilities

14.1.1 Query Languages

The ORACLE query language is SQL, a version of the family of SEQUEL languages developed by IBM. SQL is English-like, containing many "noise" words (i.e. words which are not needed in the query, but which assist the human reader, such as "from," "where," "in," etc.) The SQL parser does not require all the noise words, and many words can be abbreviated. The language is well documented, with many examples.

The ADABAS-M query language, ADASCRIPT-M, is not English-like in the sense that noise words are not used or required.

Queries to SEED use the Fortran-oriented Data Manipulation Language, which is neither user-friendly nor English-like. The commands available for user are strictly dependent on the definition of the data and relationships defined for the data base.

14.1.2 Interactive User Interface

In ORACLE, the SQL interactive processor is useful for debugging and quickly inspecting data base contents. However, it has some undesirable features:

Although SQL prints column headings for a query, the heading is truncated to the length of the field.

The data buffer size is a hindrance with the I&W data base because some records are too large to be displayed at one time. When this happens, the user is forced to select only certain fields for display. However, note that ADABAS-M does not permit the user to select fields for display, and all fields are always returned.

In ADABAS-M, the interactive debugging program was used for inspecting data base contents, as well as for debugging and other purposes.

ADABAS-M use in general is more limited by the role assigned to the Data Base Administrator (DBA). The DBA must specifically define and allow user views for the end user, open files which the user will be allowed to access, and perform other functions which restrict user access to the system.

The AMTEST routine used by ADABAS-M does not display field headings. Data are simply packed into a record buffer and returned -- all the data, all the time.

AMTEST is much more "procedural" than SQL; that is, once into AMTEST, the user must open a thread, open the file that will be accessed, put the filename into the control block, and specify the user view that will used, and its length, before a query can be made against that file. If a different file will be used, the same steps must be performed.

AMTEST allows the user to save information returned from a query to a user-specified external file, while SQL does not. In SQL, however, once the user specifies the data base name, then all files in that data base are made available.

Two interactive query facilities are provided in SEED: HARVEST and GARDEN.

HARVEST provides information about the schema/subschema definition, displays data values of specified items in the data base, and performs other functions. As in ORACLE, the user has the ability to display only selected fields.

The first step in HARVEST is to specify the subschema name and password. Once this is complete, the user is connected to the data base, and all data (files, records, sets) are available. HARVEST also allows the user to save information returned from a query in a user-specified file.

Note that HARVEST is used only to query the data base. No commands are available for the user to modify or update the data base.

GARDEN, the SEED on-line query facility, is best suited to aid in application development. GARDEN provides an interactive Data Manipulation Language (DML) which allows the programmer to obtain information about the data base (sets, records, data items), test applications programs, and, if necessary, modify the contents of the data base interactively.

An end-user, such as an analyst, would find GARDEN much too procedurally oriented. GARDEN does not provide an English-like query language and can only be used with a thorough understanding of the structure of, and relationships within, the data base.

Neither HARVEST nor GARDEN was directly evaluated during this project.

14.1.3 Host Language Interface

In general, ADABAS-M requires fewer calls to process a query. Typically, the steps performed are:

- Open a thread (connect to the data base).

- Open the file to be operated on; use the host language to set up search buffers and other required files.

- Issue the command to the data base; i.e., for a simple retrieval based on a key for an entire record, the record contents would be retrieved in a record buffer. The user is then left to "process" or "view" these field values based on relative field positions.

- Close the file.

- Close the thread (disconnect from ADABAS-M).

ORACLE requires more calls (probably because it is intended to be "user-friendly"). In general, the following commands are used:

- Log on to ORACLE.

- Open a cursor (connect to the data base).

- Issue the SQL command to the parser. (Use of the parser reflects the "user-friendly" approach, since the more natural commands require more processing.)

- Bind any search variables to the command. (The ADABAS-M equivalent would be to set up the search buffer. In comparing the two systems, notice that when using the Host Language Interface (HLI), ORACLE has the user make several calls to ORACLE-specific routines, in order to check each step, while ADABAS-M has the user make use of the host language.)

- Define receiving fields for a retrieval instead of having retrieved data dumped into a data array.

- Execute the retrieval; i.e., perform a search. However, the data will not be delivered without a FETCH command.

- FETCH the data. (In ADABAS-M the user has the option of passing a record buffer in most cases. If the buffer has been defined, the data are put into it.)

- Close the cursor (disconnect from the data base).

- Log off from ORACLE.

In SEED, the steps required to process a query are generally very simple:

- Open the data base.

- Issue the commands to the data base.

- Close the data base.

This procedure would require issuing *SEED DML* commands which are specific to the structure of the files defined in the schema. That is, a thorough knowledge of the data base definition is required for navigation.

A simple retrieval based on a key for an entire record for the Weather Summary file used in these tests would involve:

1. Finding the position in the owner set based on calc key value (a unique retrieval key used by a hashing scheme to calculate the location of the page address on which the record is placed), and then:

2. Finding members of that set based on chaining methods defined. The I&W data base has been defined with only forward pointers, and would find the next positional member of the set. The user has the option here of simply locating the record or using a variation of the same command, which will transfer the data from system buffers to the user work area.

Note that with SEED running under IAS, it was found that all applications interacting with SEED had to be taskbuilt using "split task architecture" and run from MCR (an IAS facility) as real-time tasks. Using split-task architecture, the application is linked to a vendor-supplied module which spawns a subprocess to execute the DML commands.

One feature of the SEED DML is that it includes commands which allow the user to display diagnostic messages when errors occur.

Another, less desirable feature is that, unlike ADABAS-M and ORACLE, when an error occurs in an applications program, the user must reset the error status code to zero.

For debugging purposes, it is not always necesssary (or possible) to have a task execute "successfully" or completely. Unexpected errors may occur, and the user may abort the task. When this happens in ADABAS-M, the Data Base Administrator (DBA) must use a utility to close all threads left open by the user (that is, disconnect the user from ADABAS normally). When this happens in ORACLE, the user and DBA need not take any action, because the ORACLE cleanup task detects the user task abort and frees the data base resources in use by abnormally terminating user programs and terminal processes.

Host Language Interface (HLI) documentation is always somewhat cryptic at first glance. For example, the vendors supply examples in several different languages. A problem would occur if the actual implementation failed to match specifications. The documentation might explicitly state (or even strongly imply) that certain parameters were optional in a calling sequence, but experimentation might show that they were necessary.

An example of this type of difficulty, in the ORACLE DBA documentation, occurred when project personnel used the ORACLE EXPORT utility to back up portions of the data base. Documentation indicated that certain parameters were optional and could be entered in any order. After a great deal of experimentation, it was found that all parameters were actually required, and that they must be entered in the correct order. In addition, it was found that after data files had been backed up with the EXPORT utility, they could not always be reloaded with the IMPORT utility, which sometimes indicated errors in data formats.

14.1.4 Data Base Administrator Utilities

In general, for ADABAS-M, the DBA has the ultimate power over use of the data base, and the DBA functions are quite separate and distinct from the functions that the user is allowed to perform. This would be an advantage in an environment in which the user does not want or need access to the data base at a higher level. For example, a user who wants to change a user view in ADABAS-M must call in the DBA to perform this task.

Although ADABAS-M is not a secure system, the DBA can severely limit the data base user.

ORACLE, as a more user-friendly system, allows dynamic changes to be made in the data base by the user, and gives no impression of a distinct separation in roles between the DBA and the user. ORACLE does document a privilege granting scheme.

Which of these systems is easier to use? From the point of view of the user, working interactively with the system:

- It depends on the documentation available to the user.

- It depends on how well the user knows the contents of the data base. (This is the most important factor, since SEED requires the user to understand data base structure for effective use of the system.)

- Given that the user is an analyst at a terminal, trying to process what has been seen and finding anything new based on previous information, then ORACLE would appear to be the easiest to use. It might be noted here that in ORACLE a query can be performed based on the value of *any* field, if response time is ignored. In ADABAS, a query can be performed only on a field defined as a key field, and that ADABAS might be limited in use for *ad hoc* queries unless the user had little imagination (the test is in the imagination of the data base designer). This preference for ORACLE is also based on the assumption that SQL is being compared with AMTEST, and AMTEST would require too much set-up time.

Following completion of the RTE test runs with ADABAS-M, the system vendor, Software AG, announced that the system was being withdrawn from further sales because of possible errors in the implementation. It is not clear at this time what the future status of ADABAS-M will be.

14.2 Analyst-User Viewpoint

This subsection contains, in outline form, characteristics of the three DBMSs which appear to be most significant from the point of view of the analyst-user.

- Interactive Language
    - ORACLE
        - Easiest to learn and use but only average in execution
        - English-like language
    - ADABAS-M
        - Not as English-like as ORACLE
    - SEED
        - Two interactive language options, one for each type of user:

-- HARVEST

--- Very easy for the non-technical person to use

--- On-line HELP facility to aid the user

-- GARDEN

--- Resembles host interface language

--- Requires programming background to use and understand it

o For ORACLE and ADABAS-M, the interactive language resembles the host interface language. For SEED, one interactive language, GARDEN, resembles the host interface language.

o Retrievals from non-keyed fields:

- Cannot be performed under ADABAS-M

- ORACLE and SEED can retrieve on any field, not just keyed fields. This feature is important for ad hoc query entry

o Data retrieval output facilities

- ADABAS-M

-- No field headings or separation among output items

-- Very difficult to read in this form

-- Entire record is stored in packed format in the record buffer

- ORACLE

  - -- Field headings with their values

  - -- If the heading is larger than its value, the heading is truncated. This may cause trouble if two headings have the same initial letters.

- SEED

  - -- Headings and field values

o User views: versatility of facilities for creating unique views of the data base for each user

- ORACLE

  - -- Each analyst can create his or her own table or view of the data base without changing the actual data base.

- ADABAS-M, SEED

  - -- Cannot dynamically create user views. In ADABAS-M the Data Base Administrator has a dedicated terminal and must take down the entire system to create new user views.

- Analyst aids
  - SEED
    - User-friendly on-line Help facility
  - ADABAS-M, ORACLE
    - Reference manuals required for questions
- Number of users
  - ADABAS-M
    - There is a limit on the number of files that can be open at the same time. Users can access only the files that are open at the time.
  - ORACLE
    - When the system is used interactively, the number of users that can use the system is limited. This limitation may be due to the PDP-11/70 implementation used for testing.
- Overall evaluation from point of view of user-analyst
  - ORACLE preferred from user's viewpoint
    - Able to retrieve on non-keyed fields, best for ad hoc queries

    - Easiest to identify and retrieve data

    - Provides relational joins, needed for ad hoc queries

    - Easiest to learn

  - Negative factor for ORACLE

    - Limited data buffer size. If the record is larger than the record buffer size, it cannot be displayed directly.

14.3 Programmer Viewpoint

This subsection contains evaluation factors seen from the point of view of the programmer responsible for installation and maintenance of the DBMS.

- Debugging aids

  - SEED

    - Prints description of error

  - ADABAS-M

    - Prints error code, which must be looked up in manuals

  - ORACLE

    - Prints error code only

    - Error codes cannot be predicted under certain conditions

-- Even with error codes, it is sometimes difficult to determine the actual error

-- Documentation concerning error codes less complete than ADABAS-M and SEED

o Documentation

- ADABAS-M

  -- One manual; describes host language interface.

  -- Commands given in alphabetic order

  -- No examples

  -- Well written, well organized, user is kept in mind

- ORACLE

  -- One manual

  -- Many examples

  -- Many system specific items

  -- Not well organized for programmer

  -- Understanding the manual requires that the reader know the query language

- SEED

  -- Commands listed in alphabetic order

  -- Examples too simple

  -- Structure seemed confusing

  -- Training manual was required for some information

  -- Documentation not up-to-date; missing documentation may be due to the fact that SEED was modified for this project by the vendor for use under IAS operating system.

o Record retrievals

- ADABAS-M, SEED

  -- Will return record that is being retrieved

- ORACLE

  -- Will not find and get the record at the same time; two commands are required

o Changes to data base

- SEED

  -- If the file structure changes, with fields added or deleted, the data base administrator must rebuild the system.

- ADABAS-M, ORACLE

  - Changes have minimum effect on programs; it is necessary only to modify the INCLUDE file and recompile

o Interference among users

- ADABAS-M

  - If a program aborts while holding a record, this may cause other users to abort if they are trying to access that same record. One user can interfere with other users.

- ORACLE, SEED

  - Users are independent, cannot interfere with each other

o Overall evaluation from point of view of programmer

- SEED

  - Best for quick, short retrievals; SEED's use of calc fields for data storage is extremely fast

- ADABAS-M

  - Best for complex queries on multiple key fields

- ORACLE

  -- Best compromise for efficiency in both simple and complex queries

  -- Only system to support relational joins for ad hoc queries

15.0 CONCLUSIONS AND RECOMMENDATIONS

This section presents conclusions concerning the applicability of the three DBMSs to the I&W mission. The body of the report should be consulted for further discussion (Section 1.2) and details (Sections 13.0 and 14.0).

15.1 Conclusions

ADABAS-M, with identified enhancements, is capable of supporting typical large I&W systems for the near to mid-term (+5 years).

- It can support approximately 25 I&W analysts concurrently with adequate response times (5-10 secs.) (Section 13.2).

- It has relatively efficient reorganization capabilities (Section 14.3).

ORACLE, also with identified enhancements, is capable of supporting typical small I&W systems for the near to mid-term (+5 years).

- It can support approximately five I&W analysts concurrently with adequate response times (Section 13.2).

- Its support in terms of ease of use and logical power is the greatest of the DBMSs tested (Sections 13.0 and 14.0).

- Its flexibility in terms of data base definition and reorganization is also the greatest of the DBMSs tested (Section 13.0 and 14.0).

SEED, because of its complexity and inflexibility of data base design and reorginization, is not appropriate to the I&W task, which requires great flexibility in data base structure and contents.

- Reorganization is very difficult in SEED.

- Access paths must be incorporated into the data definition, effectively ruling out the support of unpredetermined inquiry and research processes.

- SEED is the most efficient in performing simple queries which have been anticipated at data definition time, but slows down dramatically as searches become more complex.

None of the DBMSs were found to be robust. All contained program code errors which caused either invalid results or program or system aborts (Section 13.0 and 14.0).

15.2 Recommendations

Although ADABAS-M is considered to be the most efficient of the DBMSs tested, and the only one capable of supporting a large I&W implementation, it is not apparent that it would offer any advantages over currently fielded systems and Air Force owned DBMSs such as SARP V.

ADABAS-M and SEED, which was found to be inappropriate to the I&W environment, are not recommended for new I&W system development.

ORACLE is recommended for interim I&W system development where a small number of users require a powerful data model and query capability, and great flexibility in data base re-design and reorganization.

Detailed recommendations for improvement of these DBMSs can be found in Section 13.2. All would require more extensive stress testing and debugging before they could be fielded.

It is our judgement that the solution to I&W requirements for the mid to long-term (beginning approximately five years from now) will be found in currently emerging data base machine technology and future developments in special purpose function architecture.

Appendix A. DATABASE HIGHORDER INTERACTIVE LANGUAGE (DHIL)

This appendix contains definitions of the DHIL commands which were used in developing the scenarios for this project.

The full DHIL includes approximately 100 commands for scenario specification and control. This summary contains only those commands that are needed for interpreting the scenarios in Appendix C. In the definitions provided here, a brief description of the command is given, followed by a specification of its format. Explanations of the parameters are given when they are needed, and further comments describe the way in which the command has been implemented in the RTE. Since the purpose of the scenarios has been to provide a basis for evaluation of the DBMSs, several of the commands have been implemented simply as WAIT times, indicating that they involve operations which are not related to the DBMS under test.

The following parameter names appear frequently in the definitions:

userview = the subset of data upon which the analyst wishes to perform an operation

fieldname = a field name, within the userview, on which the selection is based

fieldvalue = the value associated with fieldname when making a selection request

LOGOP = optional when multiple selection criteria are required; the logical operator used to make the association between multiple fieldname/fieldvalues within a selection request (AND, OR)

RELOP = a relational operator (EQ, NE, GT, LT, GE, LE)

The following DHIL commands have been implemented in developing the scenarios:

ADDON: Add a new record to the data base.

```
        ADDON userview;
```

The assumption has been made that to add a new record to the data base, the analyst must first call up the appropriate form, ENTER all relevant information into the screen, then use the ADDON command to update the data base with this new record.

The Update scenario uses this command heavily; the translator should devise an appropriate scheme to represent the data base impact.

ALERT: Output message to analyst of the number and type of each entry in the queue.

```
        ALERT;
```

The following assumptions have been made:

Each analyst has a queue which will contain brief entries describing messages, memos, "chatters," etc., that have been put into the queue. DHIL provides a set of commands for operating on these queues.

Every message in the data base will be stored in one (and only one) file. Messages are added to this data base file by a "phantom" process called MSGIN (MSGIN will act as the "update" scenario).

The process is as follows:

1. Get message from external source.

2. Assign Information Source ID Code.

3. Add new message to data base message file.

4. Based on addressee information:

4a. Update the appropriate analysts' queues by entering the Information Source ID Code and other necessary information (priority, title, etc.)

4b. When in doubt about who sees it, update the Watch Officer's queue with the information as above.

5. Go to Step 1.

CHATTER: Transfer current display to another analyst.

```
CHATTER receiving LOGIN identifier [, receiving LOGIN
     identifier];
```

This command updates the receiving analysts' queues and writes all information to external files. The translator should generate a WAIT time.

COMMENT: Scenario remarks.

```
COMMENT scenario-comments;
```

Allows the insertion of comments to give detail on how or why certain steps will be performed in the scenario. The translator has the option of ignoring the comment command, or reproducing it within the script.

DELETE: Delete the first record that meets the selection criteria from the data base.

```
DELETE ([userview, fieldname RELOP fieldvalue
     [LOGOP fieldname RELOP fieldvalue]]);
```

DELETE without a parameter list (i.e. DELETE();) will cause the removal of the currently accessed record from the data base.

DELETE with a parameter list causes the translator to find the record that meets the selection criteria, then remove it from the data base (actually a RETRIEVE (parameter list); followed by a DELETE();).

userview and fieldname are standard

fieldvalue = any valid field value (constant, character string, etc.)

If no selection criterion is specified, delete the current record.

DISPLAY: Display full body of alert.

```
DISPLAY [<alert number>][into form <userview>];
```

This command represents the analyst displaying something in the queue other than a message, because a message retrieval represents a data base access, while all other types of alerts should be stored in external files. The translator could set up an arbitrary retrieval from a non-database file, or set up a WAIT time.

If <alert number> is not specified, default to highest priority alert in queue. "DISPLAY [<alert number>] into form <userview>" is used to fill a data base record format with the data contained in the specified alert queue entry.

ENTER: Used in the RTE to simulate the analyst entering data into a previously retrieved form (message form, collection request form, etc.)

```
ENTER (fieldname = newvalue [, fieldname = newvalue]);
```

Only the contents of the display are modified; this command has no effect on the data base. Subsequent primitive directives control the disposition of the data. The translator should handle this command as a WAIT time.

FORM: Display the template format for the specified user view.

```
FORM userview;
```

This command is used to display a form which will then have information ENTERed into it by the analyst (a sequence of steps used to add a record to the data base). The translator should show activity against whatever files this FORM is contained in, or, if this is impossible, simply execute a WAIT time.

HARDCOPY: Format and transfer all of current display to hardcopy device.

```
HARDCOPY;
```

The translator should handle this command as a WAIT time.

INSERT: Add DBMS items (values) to graphics display and insert information about those items into data blocks.

```
INSERT ([class = <symbol>],[loc = <value>], block = <value>);
```

Parameters are:

class optional parameter used to assign a symbol to an item being added to display.

loc optional parameter: the coordinates of the item being added to display. If not specified, default to current position of cursor or light pen.

block required parameter, used to append a data block to the item identified.

The INSERT command is issued as a WAIT time from the script.

MAP: Generate map background.

```
MAP ([LAT = <fieldvalue>, LON = <fieldvalue> / LOC = <location
    name>], [SCALE = <scale>]);
```

The analyst must specify either LAT and LON, or LOC, but not both.

LAT, LON = the geographic coordinates to be centered on display

LOC = country/region/major city/installation to be centered on display

SCALE = optional parameter which specifies the desired map scale. If not specified default to the largest scale (1:20m). Scales available are: 1:20m, 1:5m, 1:1m, 1:500k, 1:250k, 1:100k, 1:50k, 1:25k, 1:10k, 1:7.5k.

The translator should make the appropriate retrievals against whatever data base file this information is stored in.

MODIFY: Change current selected data base items (single instance)

```
MODIFY (fieldname = new value [, fieldname = new value]);
```

fieldname = the name(s) of the field(s) within the record that should be modified.

new value = any valid field value (constant, character string, variable, etc.)

This command is used to represent the analyst making an update to the specified fields in the currently accessed record. The translator should update the specified record in the data base.

PAGE: Scroll the display image forward or backward one "page" at a time. This command is used for the RTE to simulate the ai.alyst pressing the appropriate function key for scrolling.

```
PAGE;
```

The translator should handle this command as a WAIT time.

PERFORM: Execute a function and return.

PERFORM function/procedure/subroutine name;

The translator should replace the PERFORM command with the appropriate FORTRAN CALL statement.

PLOT: Retrieve DBMS items and display on previously generated map.

PLOT ([userview, fieldname RELOP fieldvalue [LOGOP fieldname RELOP fieldvalue]], CLASS = symbol, [BLOCK/NOBLOCK]);

Retrieve multiple instances of data that qualify by the specified selection criteria and plot onto current map display.

CLASS = classification of symbolic data to be presented.

BLOCK/NOBLOCK = optionally specifies that a data block/tag is to be appended to each symbol generated for display. (BLOCK is the default.)

The translator should generate commands for the necessary data base search/retrieval based on the selection criteria specified.

PLOT without selection criteria (PLOT (, CLASS = symbol, [BLOCK/NOBLOCK]);) allows the analyst to plot the information associated with the record currently being displayed. The translator handles this as a WAIT time.

QUERY: Select specified data (multiple instances - list display).

QUERY (userview, fieldname RELOP fieldvalue [LOGOP fieldname RELOP fieldvalue], [SORT BY fieldname [ASC/DESC]]);

The parameter list follows the standard definition.

QUERY will retrieve all records in the data base that meet the selection criteria and display them sequentially to the user in SORTed order (if specified) with scroll capability. The translator should perform the steps required to access the specified records, and perform the SORT (if specified) on those records found. (For the RTE, SCROLL is represented by the PAGE command.)

RECALL: Return the most recently STOREd display (from the top of the STORE file) to the terminal.

```
     RECALL [STORE];
```

If the optional parameter STORE is given, the current display will be STOREd after the top display is RECALLed.

The translator handles the RECALL by reading the top "display" (most recently stored display) from the STORE file, displaying it to the user, and removing this information from the STORE file. In the RTE, the second step (displaying the information to the user) is omitted.

If the STORE parameter is given, move the current display information to a temporary file, perform a RECALL, then STORE the information from the temporary file to the STORE file.

RETRIEVE: Retrieves the first occurrence of a record that meets the selection criteria (from userview), and displays it to the analyst through the userview.

```
     RETRIEVE (userview, fieldname RELOP fieldvalue [LOGOP fieldname RELOP
fieldvalue]);
```

The translator should perform the steps required to access the specified data base record.

REVIEW: Retrieve and display next record of data that qualify by the selection criteria in the previous SEARCH statement (get next record (in SORTed order) from hit-list). When end of data is reached transfer control to statement following END-SEARCH.

```
     REVIEW;
```

The translator will handle the SEARCH/REVIEW pair according to the types of data base access commands available for each DBMS.

ROUTE: The ROUTE command is used to notify an analyst (<receiving login identifier>) that the specified message should be seen by inserting the information associated with that message into the alert queue.

```
     ROUTE (LOGIN-ID = <receiving login identifier>, MSG-ID =
<fieldvalue>, PRI = <fieldvalue>, SUBJ = <fieldvalue>);
```

The ROUTE command is issued as a WAIT time from the script.

SCAN: Display short titles of all queued alerts.

```
     SCAN;
```

Since these queues will not be implemented for the RTE, the translator can substitute a WAIT time.

SEG: Draw a line segment between two points.

```
     SEG([FROM-LAT = <fieldvalue>, FROM-LON = <fieldvalue>], TO-LAT =
<fieldvalue>, TO-LON = <fieldvalue>);
```

Notice that the FROM coordinates are optional parameters. If not specified, the assumption is that the analyst has identified the FROM coordinates with the current position of the cursor or light pen.

The translator executes a WAIT time for the RTE.

SOMETIMES: Used to support the RTE by "randomly" determining (using a pseudo-random number genrator) whether or not the associated set of commands should be executed.

```
SOMETIMES (probability, [A], [LOOP = <integer>]);
```

probability = a real number (hard-coded into the scenario) which assigns the probability of some "event" occurring, i.e. that the commands contained within the following BEG-SOME and END-SOME delimiters will be executed.

A = optional parameter which when present indicates that the alert status should weight the probability. For the RTE, it is assumed that:

PEACE implies that probability = probability

CRISIS implies that probability = probability * 2 or probability = 1, whichever is less.

WAR implies that probability = probability * 3 or probability = 1, whichever is less.

LOOP = <integer> is an optional parameter which when present indicates that the SOMETIMES command should be repeated <integer> times, with the probability weighting each iteration. The default is a single iteration.

This approach always presumes that a "higher" alert status increases the probability of all events occurring.

SOMETIMES is used with the BEG-SOME and END-SOME primitives (for compound statements). The translator should set up a SOMETIMES function which, when called, will calculate the "overall" probability (probability * A) and use a random number generator to determine if the event should occur, so that translation should produce:

```
    IF (SOMETIMES (PROB, A)) THEN          (*sometimes()*)
         BEGIN                             (*beg-some*)
         ...                                 ...
         END                               (*end-some*)
```

STORE: Save the current display for later RECALL. The STORE/RECALL primitives act as stack operators (PUSH and POP respectively to a temporary file) always acting on the top of the stack/file (last in, first out).

```
    STORE;
```

The translator should handle this command by writing the current display information to the top of a temporary store-file.

Up to three displays may be STOREd at one time; if an analyst attempts to STORE a fourth display, delete the bottom display from the STORE file, then STORE the new display.

THINK: Simulates "think-time" by generating a random number of seconds to WAIT.

THINK (min, max, [A]);

min = minimum number of seconds to WAIT

max = maximum number of seconds to WAIT

A = optional parameter, representing alert status, which when specified weights min and max as follows:

PEACE implies min = min, max = max.

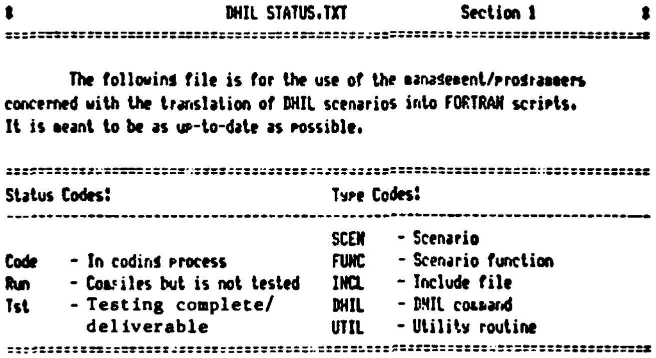
CRISIS implies min = int(min/2), max = int(max/2)

WAR implies min = int(min/3), max = int(max/3)

This command is used to represent the time lapse which occurs during a scenario when an analyst reads something, waits for hard copy, decides what to do next, takes a coffee break, etc. Use of the A parameter simulates the effect that alert status has on an analyst's activity.

Table A-1 contains detailed instructions for the translation of DHIL statements into FORTRAN script elements.

To ensure adequate managerial visibility into the progress of code translation, testing and implementation, and to act as a common communication medium for the different translators, a module-by-module tracking system was implemented. As each module would flow through the cycle from module design to final acceptance of the code, its current status would be indicated in an on-line report file.



```
*                     DHIL STATUS.TXT          Section 1          *
===================================================================

      The following file is for the use of the management/programmers
concerned with the translation of DHIL scenarios into FORTRAN scripts.
It is meant to be as up-to-date as possible.

===================================================================
Status Codes:                        Type Codes:
-------------------------------------------------------------------
                                     SCEN    - Scenario
Code    - In coding process          FUNC    - Scenario function
Run     - Compiles but is not tested INCL    - Include file
Tst     - Testing complete/          DHIL    - DHIL command
          deliverable                UTIL    - Utility routine
===================================================================
```

Table A-1 Translation of DHIL Statements (page 1 of 9)

```
===============================================================================
*                          DHIL STATUS.TXT            Section 2               *
===============================================================================

DBMS:                   ADABAS - M DEPENDENT

Mod-Nam Description    Type    Date    Author  Status  Comment
--------+--------------+-------+-------+-------+-------+---------------------
WATCH   | WATCH-FUNC   | SCEN  | 9/27  |[200,1]| Tst   | SCENARIO COMPLETE !!!
--------+--------------+-------+-------+-------+-------+---------------------
REVINP  | REVIEW-INPUTS| FUNC  | 9/16  |[200,1]| Tst   |IW,AA
--------+--------------+-------+-------+-------+-------+---------------------
CMPMSG  | COMP-MSG-HOLD| FUNC  | 9/17  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
RVEVNT  | REVIEW-EVENTS| FUNC  | 9/23  |[200,1]| Tst   | ! opens second thread
--------+--------------+-------+-------+-------+-------+---------------------
EXSIND  | EXAM-SIGN-IND| FUNC  | 9/23  |[200,1]| Tst   | ! opens second thread
--------+--------------+-------+-------+-------+-------+---------------------
GETOPS  | GET-CRNT-OPS | FUNC  | 9/22  |[200,1]| Tst   | ! opens second thread
--------+--------------+-------+-------+-------+-------+---------------------
PRWMSG  | PREP-WARN-MSG| FUNC  | 9/23  |[200,1]| Tst   | !TIME for unique key
 --adds records to all A-files, then deletes them before returning to mainline
--------+--------------+-------+-------+-------+-------+---------------------
UPDFIL  | UPDATE-FILES | FUNC  | 9/22  |[200,1]| Tst   | ! updates E001A
--------+--------------+-------+-------+-------+-------+---------------------
        | AREA-ANALYSIS SCENARIO       |
--------+--------------+-------+-------+-------+-------+---------------------
AREA    | AREA-ANALYSIS| SCRIPT| 9/30  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
CMPHLD  | COMP-HOLDINGS| FUNC  | 9/24  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
UPDTOB  | UPDATE-OB    | FUNC  | 9/17  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
ADLDAT  | ADDL-DATA-REQD| FUNC | 9/30  |[200,1]| Tst   |! opens second thread
--------+--------------+-------+-------+-------+-------+---------------------
GETCAP  | GET-CAPABILITY| FUNC | 9/24  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
ACTIND  | ACTIVE-INDICA| FUNC  | 9/20  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
COLREQ  | COLL-REQ     | FUNC  | 9/29  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
PLTRTE  | PLOT-FLT-ROUTE| FUNC | 9/29  |[200,1]| Tst   |IW
--------+--------------+-------+-------+-------+-------+---------------------
MSNUPD  | MISSION-UPDATE| FUNC | 9/28  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
EXMIND  | EXAMINE-INDICA| FUNC | 9/27  |[200,1]| Tst   |
--------+--------------+-------+-------+-------+-------+---------------------
PREMSG  | PREPARE-MSG  | FUNC  | 9/28  |[200,1]| Tst   |!same as PRWMSG above
--------+--------------+-------+-------+-------+-------+---------------------
        | IW-ANALYSIS SCENARIO         |
--------+--------------+-------+-------+-------+-------+---------------------
IWANAL  | IW-ANAYLSIS  | Script| 7/12  |       | Code  |
--------+--------------+-------+-------+-------+-------+---------------------
FNDACT  | FIND-ALL-ACTIV| FUNC | 7/12  |       | Code  |
--------+--------------+-------+-------+-------+-------+---------------------
BNDARE  | BOUND-AREA   | FUNC  | 7/12  |       | Code  |
--------+--------------+-------+-------+-------+-------+---------------------
PLTAOB  | PLOT-AOB     | FUNC  | 7/12  |       | Code  |
--------+--------------+-------+-------+-------+-------+---------------------
CONTIM  | COMPUTE-TIME | FUNC  | 7/12  |       | Code  |
```

Table A-1 Translation of DHIL Statements (page 2 of 9)

BLDMAP	BUILD-AREA-MAP	FUNC	7/15		Code	IMS
NOTID	NOT-IDENTIFIED	FUNC	7/14		Code	
	MAC-ROUTE-ASSESSMENT					
MACRTE	MAC-ROUTE	SCRIPT	7/20		Code	
PLTMSN	PLOT-MSN	FUNC	7/13		Code	
CONINF	COUNTRY-INFO	FUNC	7/15		Code	
SRCHOB	SEARCH-OB	FUNC	7/13		Code	
FNDCAP	FIND-SYS-CAP	FUNC	7/13		Code	
SRCHEV	SEARCH-EVENTS	FUNC	7/15		Code	
QRYIND	QUERY-INDICATOR	FUNC	7/15		Code	
	MIL-SYS-ANAL SCENARIO					
MILSYS	MIL-SYS-ANAL	SCEN	7/12		Code	
UPSYSD	UPDATE-SYS-DAT	FUNC	7/26		Run	
UPDSAM	UPDAT SAM INFO	FUNC	7/26		Run	
PLOTSS	PLOT-SAM-SITES	FUNC	7/26		Run	
PLOTRS	PLOT-RADAR-STS	FUNC	7/26		Run	
OVMSNS	OVERLAY-MSNS	FUNC	7/12		Run	
UPDRI	UPDAT-RADAR-IN	FUNC	7/26		Run	
SYSCOL	SYS-COLL-REQ	FUNC	7/29		Run	
	COLLECTION-COORDINATION SCENARIO					
COLCOR	COLLECTION-COR	SCEN	9/29	[200,1]	Tst	
GETCA	GET-COL-AUTHOR	FUNC	6/28	[200,1]	Tst	
REJECT	REJECT-REQUEST	FUNC	6/28	[200,1]	Tst	
REROUT	REROUTE-REQUST	FUNC	6/28	[200,1]	Tst	
EXTREQ	EXTRACT-REQMNT	FUNC	9/29	[200,1]	Tst	
REVRCE	REVIEW-RECCE	FUNC	9/29	[200,1]	Tst	
	A SPECIAL COLL-COORD SCENARIO PAIR:					
ADDTOG	ADD-TO-G001A	SCEN	9/29	[200,1]	Tst	! adds 50 requests
COLDEL	COLL-COORD-DEL	SCEN	9/29	[200,1]	Tst	! deletes requests after processing
EXTDEL	EXTREQ W/DELETE	FUNC	9/29	[200,1]	Tst	!same as extract-and-process-requirements (above), except that this version deletes the requests once they have been processed ...
REJCTH	REJECT(W/HOLD)	FUNC	9/29	[200,1]	Tst	! REJECT, but hold rec

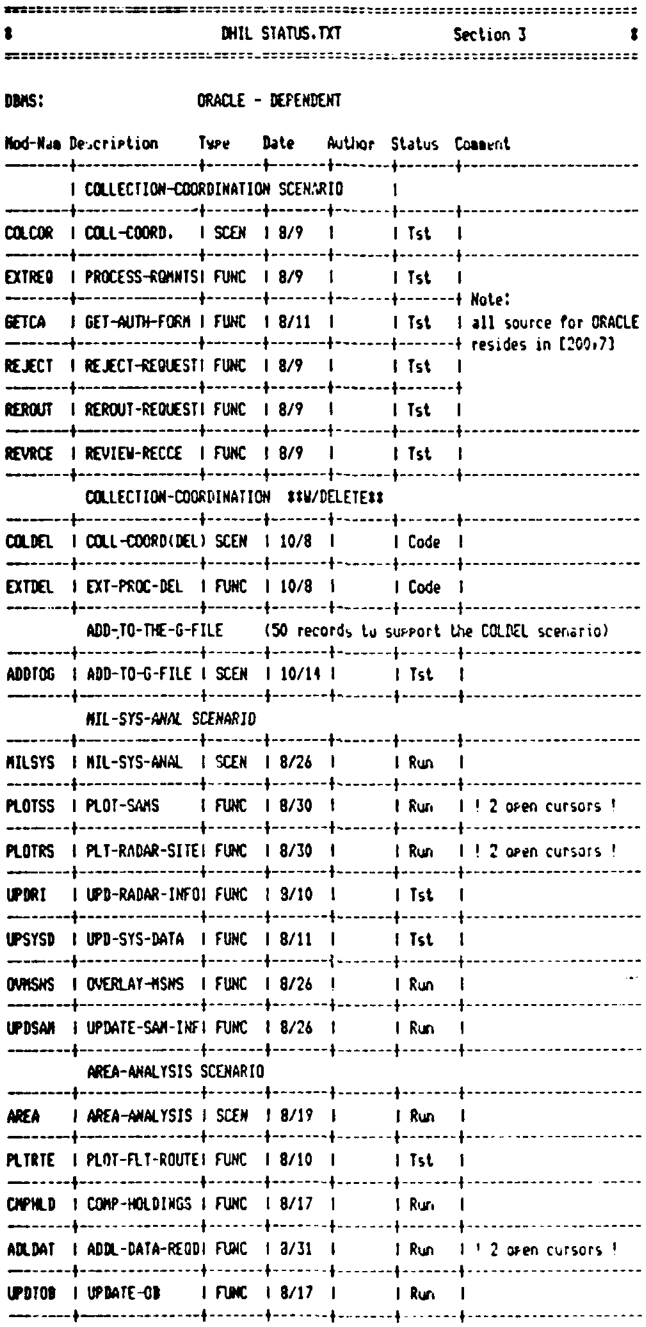
Table A-1 Translation of [illegible] Statements (pa[illegible] of 9)

```
RERTEH  | REROUT(W/HOLD)| FUNC  | 9/29  |[200,1]| Tst   |' REROUT, but hold rec
--------+---------------+-------+-------+-------+-------+---------------------
FOR THE VERSION OF THE COLLECTION-COORDINATION SCENARIO THAT UPDATES RECORDS
AS THEY ARE PROCESSED, LINK [200,1]COLDEL WITH [200,15]DHIL/LB.
--------+---------------+-------+-------+-------+-------+---------------------
        | UPDATE SCENARIO       |
--------+---------------+-------+-------+-------+-------+---------------------
UPDATE  | UPDATE        | SCEN  | 7/20  |       | Code  |
--------+---------------+-------+-------+-------+-------+---------------------
ADDMSG  | ADD-MSG-TO-DB | FUNC  | 7/19  |       | Code  |
--------+---------------+-------+-------+-------+-------+---------------------
DELETE  | DELETE MSGS   | FUNC  | 7/21  |       | Run   |
--------+---------------+-------+-------+-------+-------+---------------------



SCI> ^R

--------+---------------+-------+-------+-------+-------+---------------------
ADACOM  | ADABAS Common | INCL  | 6/2   | JWT   | Tst   | Adabas communication
--------+---------------+-------+-------+-------+-------+---------------------
ADATWO  | 2nd CTRLBK    | INCL  | 7/19  |       | Code  | for circle searches
--------+---------------+-------+-------+-------+-------+---------------------
```

SIMPLE - multiple (100) simple retrievals against the "J-File" (weather info.)
based on a random single key value (country code).

COMPLEX - multiple (100) complex queries against the "J-File" based on
four randomly generated key values (place name 1 and country code 1 OR
place name 2 and country code 2).

Table A-1 Translation of DHIL Statements (page 4 of 9)



```
============================================================================
*                          DHIL STATUS.TXT          Section 3              *
============================================================================

DBMS:                  ORACLE - DEPENDENT

Mod-Nam Description    Type   Date   Author Status Comment
--------+--------------+------+------+------+------+--------------------
        | COLLECTION-COORDINATION SCENARIO   |
--------+--------------+------+------+------+------+--------------------
COLCOR  | COLL-COORD.  | SCEN | 8/9  |      | Tst  |
--------+--------------+------+------+------+------+--------------------
EXTREQ  | PROCESS-RQMNTS| FUNC | 8/9  |      | Tst  |
--------+--------------+------+------+------+------+ Note:
GETCA   | GET-AUTH-FORM | FUNC | 8/11 |      | Tst  | all source for ORACLE
--------+--------------+------+------+------+------+ resides in [200,7]
REJECT  | REJECT-REQUEST| FUNC | 8/9  |      | Tst  |
--------+--------------+------+------+------+------+
REROUT  | REROUT-REQUEST| FUNC | 8/9  |      | Tst  |
--------+--------------+------+------+------+------+--------------------
REVRCE  | REVIEW-RECCE | FUNC | 8/9  |      | Tst  |
--------+--------------+------+------+------+------+--------------------
          COLLECTION-COORDINATION  **W/DELETE**
--------+--------------+------+------+------+------+--------------------
COLDEL  | COLL-COORD(DEL) SCEN | 10/8 |      | Code |
--------+--------------+------+------+------+------+--------------------
EXTDEL  | EXT-PROC-DEL | FUNC | 10/8 |      | Code |
--------+--------------+------+------+------+------+--------------------
          ADD-TO-THE-G-FILE     (50 records to support the COLDEL scenario)
--------+--------------+------+------+------+------+--------------------
ADDTOG  | ADD-TO-G-FILE | SCEN | 10/14 |     | Tst  |
--------+--------------+------+------+------+------+--------------------
          MIL-SYS-ANAL SCENARIO
--------+--------------+------+------+------+------+--------------------
MILSYS  | MIL-SYS-ANAL | SCEN | 8/26 |      | Run  |
--------+--------------+------+------+------+------+--------------------
PLOTSS  | PLOT-SAMS    | FUNC | 8/30 |      | Run  | ! 2 open cursors !
--------+--------------+------+------+------+------+--------------------
PLOTRS  | PLT-RADAR-SITE| FUNC | 8/30 |      | Run  | ! 2 open cursors !
--------+--------------+------+------+------+------+--------------------
UPDRI   | UPD-RADAR-INFO| FUNC | 9/10 |      | Tst  |
--------+--------------+------+------+------+------+--------------------
UPSYSD  | UPD-SYS-DATA | FUNC | 8/11 |      | Tst  |
--------+--------------+------+------+------+------+--------------------
OVMSMS  | OVERLAY-MSNS | FUNC | 8/26 |      | Run  |
--------+--------------+------+------+------+------+--------------------
UPDSAM  | UPDATE-SAM-INF| FUNC | 8/26 |      | Run  |
--------+--------------+------+------+------+------+--------------------
          AREA-ANALYSIS SCENARIO
--------+--------------+------+------+------+------+--------------------
AREA    | AREA-ANALYSIS | SCEN | 8/19 |      | Run  |
--------+--------------+------+------+------+------+--------------------
PLTRTE  | PLOT-FLT-ROUTE| FUNC | 8/10 |      | Tst  |
--------+--------------+------+------+------+------+--------------------
CMPHLD  | COMP-HOLDINGS | FUNC | 8/17 |      | Run  |
--------+--------------+------+------+------+------+--------------------
ADLDAT  | ADDL-DATA-REQD| FUNC | 8/31 |      | Run  | ! 2 open cursors !
--------+--------------+------+------+------+------+--------------------
UPDTOB  | UPDATE-OB    | FUNC | 8/17 |      | Run  |
--------+--------------+------+------+------+------+--------------------
```

Table A-1 Translation of DHIL Statements (page 5 of 9)

```
GETCAP  | GET-CAPABLTIES| FUNC  | 8/18  |       | Run   |
ACTIND  | ACTIVATE-IND  | FUNC  | 8/18  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
COLREQ  | COLL-REQ      | FUNC  | 8/18  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
MSNUPD  | MISSION-UPDATE| FUNC  | 8/18  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
EXMIND  | EXAM-INDICATOR| FUNC  | 8/18  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
         WATCH-FUNCTION SCENARIO
--------+---------------+-------+-------+-------+-------+--------------------
WATCH   | WATCH-FUNCTION| SCEN  | 9/1   |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
REVINP  | REVIEW-INPUTS | FUNC  | 8/19  |       | Run   | also used by AREA
--------+---------------+-------+-------+-------+-------+--------------------
CMPMSG  | COMP-MSG-HOLD | FUNC  | 8/24  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
RVEVNT  | REVIEW-EVENTS | FUNC  | 8/25  |       | Code  |
--------+---------------+-------+-------+-------+-------+--------------------
EXSIND  | EXAM-SIGNF-IND| FUNC  | 8/26  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
GETOPS  | GET-CURRENT-OP| FUNC  | 8/27  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
UPDFIL  | UPDATE-FILES  | FUNC  | 8/27  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
         MAC-ROUTE-ACCESSMENT SCENARIO
--------+---------------+-------+-------+-------+-------+--------------------
MACRTE  | MAC-ROUTE     | SCEN  | 9/1   |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
PLTMSN  | PLOT-MSN      | FUNC  | 8/30  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
CONINF  | COUNTRY-INFO  | FUNC  | 8/30  |       | Run   |
--------+---------- ----+-------+-------+-------+-------+--------------------
SRCHOB  | SEARCH-OB     | FUNC  | 8/30  |       | Run   | ! 2 open cursors !
--------+---------------+-------+-------+-------+-------+--------------------
FNDCAP  | FIND-SYS-CAP  | FUNC  | 8/30  |       | Run   |
--------+---------------+-------+-------+-------+-------+--------------------
QRYIND  | QUERY-INDICAT | FUNC  | 8/31  |       | Run   | ! 2 open cursors !
--------+---------------+-------+-------+-------+-------+--------------------
SRCHEV  | SEARCH-EVENTS | FUNC  | 8/31  |       | Run   | ! 2 open cursors !
--------+---------------+-------+-------+-------+-------+--------------------
         IW-ANALYSIS SCENARIO
--------+---------------+-------+-------+-------+-------+--------------------
FNDACT  | FIND-ALL-ACTVTY FUNC  | 9/1   |       | Run   | ! 2 open cursors !
--------+---------------+-------+-------+-------+-------+--------------------
BNDARE  | BOUND-AREA    | FUNC  | 9/2   |       | Run   | ! 2 open cursors !
--------+---------------+-------+-------+-------+-------+--------------------
PLTAOB  | PLOT-AOB      | FUNC  | 9/2   |       | Run   | ! 2 open cursors !
--------+---------------+-------+-------+-------+-------+--------------------


SCI> ^R
--------+---------------+-------+-------+-------+-------+--------------------
A1UV1   | A1.1 userview | INCL  | 8/19  | field defns. for retrievals on A001.1
--------+---------------+-------+-------+-------+-------+--------------------
B1UV1   | B1.1 userview | INCL  | 8/25  |  same, but for userview B001.1
--------+---------------+-------+-------+-------+-------+--------------------
C1UV1   | C1.1 userview | INCL  | 8/17  |  same, but for userview C001.1
--------+---------------+-------+ ------+-------+-- ---+--------------------
D1UV1   | D1.1 userview | INCL  | 8/9   |  same, but for userview D001.1
--------+---------------+-------+-------+-------+-- ---+--------------------
D2UV2   | D2.2 userview | INCL  | 8/25  |  same, but for userview D002.2
--------+---------------+-------+-------+-------+-------+- ---   ------- -
E1UV1   | E1.1 userview | INCL  | 8/25  |  same, but for userview E001.1
```

Table A-1 Translation of [illegible] Statements (page 6 of 9)

```
--------+---------------+-------+-------+-------+-------+---------------------
E2UV1   | E2.1 userview | INCL  | 8/16  |  same, but for userview E002.1
--------+---------------+-------+-------+-------+-------+---------------------
E2UV3   | E2.3 userview | INCL  | 8/16  |  same, but for userview E002.3
--------+---------------+-------+-------+-------+-------+---------------------
F1UV1   | F1.1 userview | INCL  | 8/18  |  same, but for userview F001.1
--------+---------------+-------+-------+-------+-------+---------------------
G1UV1   | G1.1 userview | INCL  | 8/3   |  same, but for userview G001.1
--------+---------------+-------+-------+-------+-------+---------------------
G1UV2   | G1.2 userview | INCL  | 8/3   |  same, but for userview G001.2
--------+---------------+-------+-------+-------+-------+---------------------
H1UV1   | H1.1 userview | INCL  | 8/3   |  same, but for userview H001.1
--------+---------------+-------+-------+-------+-------+---------------------
I2UV1   | I2.1 userview | INCL  | 8/4   |  same, but for userview I002.1
--------+---------------+-------+-------+-------+-------+---------------------
J1UV1   | J1.1 userview | INCL  | 8/3   |  same, but for userview J001.1
--------+---------------+-------+-------+-------+-------+---------------------
```

SIMPLE - multiple (100) simple retrievals against the "J-File" (weather info.) based on a random single key value (country code).

COMPLEX - multiple (100) complex queries against the "J-File" based on four randomly generated key values (place name 1 and country code 1 OR place name 2 and country code 2).

```
==============================================================================
*                          DHIL STATUS.TXT             Section 4             *
==============================================================================

DBMS:                    SEED - DEPENDENT

Mod-Nam Description      Type    Date    Author  Status  Comment
--------+---------------+-------+-------+-------+-------+---------------------
```

SIMPLE1 - multiple (100) simple retrievals against the "J-File" (weather info.) based on a random single key value (country code).

SIMPLE2 - multiple (100) simple queries against the "J-File" (weather info.) based on a random single key value (place name).

COMPLEX1 - multiple (100) complex queries against the "J-File" based on four randomly generated key values (ccode 1 and place 1 OR ccode 2 and place 2) with access via the country code set.

COMPLEX2 - multiple (100) complex queries against the "J-File" based on four randomly generated key values (ccode 1 and place 2 OR ccode 2 and place 2) with access via the place name set.

COMPLEX with UPDATE - the complex queries were performed against the "J-File" and the records found which met the selection criteria were updated.

Table A-1 Translation of DHIL Statements (page 7 of 9)

```
SCI> ^R
============================================================================
*                         DHIL STATUS.TXT          Section 5               *
============================================================================

DBMS:                   ANY

Mod-Nam Description     Type    Date    Author  Status  Comment
--------+---------------+-------+-------+-------+-------+---------------------
ALERT   | ALERT         | DHIL  | 6/7   | JWT   | Tst   |
--------+---------------+-------+-------+-------+-------+---------------------
PAGE    | PAGE          | DHIL  | 5/21  | JWT   | Tst   |
--------+---------------+-------+-------+-------+-------+---------------------
ROUTE   | ROUTE         | DHIL  | 7/23  |       | Tst   | wait 3 to 7 secs.
--------+---------------+-------+-------+-------+-------+---------------------
SCAN    | SCAN          | DHIL  | 5/24  | JWT   | Tst   |
--------+---------------+-------+-------+-------+-------+---------------------
STORE   | STORE         | DHIL  | 6/28  |       | Tst   | wait 2 to 5 secs.
--------+---------------+-------+-------+-------+-------+---------------------
THINK   | THINK         | DHIL  | 5/24  | JWT   | Tst   |
--------+---------------+-------+-------+-------+-------+---------------------
FORM    | FORM          | DHIL  | 7/29  |       | Tst   | wait 2 to 7 secs.
--------+---------------+-------+-------+-------+-------+---------------------
HRDCPY  | HARDCOPY      | DHIL  | 8/18  |       | Tst   | wait 2 to 5 secs.
--------+---------------+-------+-------+-------+-------+---------------------
RTEMSG  | ROUTE-MSG     | FUNC  | 7/14  |       | Tst   |AA,WATCH
--------+---------------+-------+-------+-------+-------+---------------------
STOMSG  | STORE-MSG     | FUNC  | 7/16  |       | Tst   |AA,WATCH
--------+---------------+-------+-------+-------+-------+---------------------
STOMAP  | STORE-MAP     | FUNC  | 7/16  |       | Tst   |AA,WATCH
--------+---------------+-------+-------+-------+-------+---------------------
EQUALA  | String Compar | UTIL  | 6/7   | JWT   | Tst   |
--------+---------------+-------+-------+-------+-------+---------------------
RND     | Random Number | UTIL  | 6/2   | DECUS | Tst   | Also used in BMD, DBG
--------+---------------+-------+-------+-------+-------+---------------------
CMPRS   | Compress      | UTIL  | 6/3   | JWT   | Tst   | Spaces, Tabs, Nulls
--------+---------------+-------+-------+-------+-------+---------------------
EVEN    | Evens integer | UTIL  | 6/7   | JWT   | Tst   | Adds one to odds
--------+---------------+-------+-------+-------+-------+---------------------
ANDLST  | 'AND' 2 lists | UTIL  | 6/3   | JWT   | Tst   |
--------+---------------+-------+-------+-------+-------+---------------------
CMPLST  | Compress list | UTIL  | 6/3   | JWT   | Run   | Get rid of blanks
--------+---------------+-------+-------+-------+-------+---------------------
DELLST  | Delete entry  | UTIL  | 6/3   | JWT   | Run   | Delete a value
--------+---------------+-------+-------+-------+-------+---------------------
FRELST  | Frees lists   | UTIL  | 6/3   | JWT   | Run   | Frees up lists
--------+---------------+-------+-------+-------+-------+---------------------
GETLST  | Get values    | UTIL  | 6/3   | JWT   | Run   | Get values from list
--------+---------------+-------+-------+-------+-------+---------------------
GIVLST  | Gives a list  | UTIL  | 6/3   | JWT   | Run   | Gives a hit list
--------+---------------+-------+-------+-------+-------+---------------------
OPNLST  | Open a list   | UTIL  | 6/3   | JWT   | Run   | **DOES NOT WORK** RGP
--------+---------------+-------+-------+-------+-------+---------------------
ORLST   | 'OR' 2 lists  | UTIL  | 6/3   | JWT   | Run   |
--------+---------------+-------+-------+-------+-------+---------------------
PUTLST  | Enters list   | UTIL  | 6/3   | JWT   | Run   | Adds element to list
--------+---------------+-------+-------+-------+-------+---------------------
ENTER   | ENTER(...)    | DHIL  | 7/1   |       | Tst   | wait 5 to 15 secs.
--------+---------------+-------+-------+-------+-------+---------------------
```

Table A-1 Translation of DHIL Statements (page 8 of 9)

CIRCLE	CIRCLE(...)	DHIL	7/1		Tst	wait 1 or 2 secs.
INSERT	INSERT(...)	DHIL	7/1		Tst	wait 2 to 5 secs.
AMP	AMP	DHIL	7/1		Tst	wait 2 to 5 secs.
SEG	SEG(...)	DHIL	7/1		Tst	wait 1 or 2 secs.
CHTTER	CHATTER ...	DHIL	7/1		Tst	wait 1 or 2 secs.
DISPLY	DISPLAY ...	DHIL	7/1		Tst	wait 1 to 3 secs.
PURGE	PURGE ...	DHIL	7/1		Tst	wait 1 or 2 secs.
RECALL	RECALL	DHIL	6/28		Tst	wait 2 to 5 secs.
CHKOS	CHECK SOURCES	FUNC	7/2		Tst	used by MIL-SYS-ANAL
MAP	MAP(...)	DHIL	7/2		Tst	wait 2 to 5 secs.
PLOT	PLOT(...)	DHIL	7/2		Tst	wait 2 to 5 secs.
****** used only by PLOT commands without a selection list ******						
PLTNSA	PLOT-NEW-SAM	FUNC	7/6		Tst	used by MIL-SYS-ANAL
CHTNSA	CHAT-NEW-SAM	FUNC	7/6		Tst	used by MIL-SYS-ANAL
CHTNSI	CHAT-NEW-SITE	FUNC	7/6		Tst	used by MIL-SYS-ANAL
CHATIC	CHAT-INC-CAP	FUNC	7/6		Tst	used by MIL-SYS-ANAL
PLOTIC	PLOT-INC-CAP	FUNC	7/6		Tst	used by MIL-SYS-ANAL
PLTNSI	PLOT-NEW-SITE	FUNC	7/6		Tst	used by MIL-SYS-ANAL
BUILDM	BUILD-MAP	FUNC	7/14		Tst	AA,WATCH
BRFTXT	BRIEF-TXT	FUNC	7/15		Tst	IW,WATCH
CHTCOL	CHATTER-COORD	FUNC	7/9		Code	used by AREA
CNTMAC	CHATTER-MAC	FUNC	7/9		Tst	used by AREA, IW
PREBRF	PRESENT-BRIEF	FUNC	6/17		Code	used by AREA
MSGIF	MSG-INTO-FORM	FUNC	7/19		Tst	used by UPDATE
RANLST	RTE-TO-ANALYST	FUNC	7/19		Tst	used by UPDATE
RWATCH	RTE-TO-WATCH	FUNC	7/19		Tst	used by UPDATE
REMOVE	REMOVE-MESSAGE	FUNC	7/19		Tst	used by UPDATE
GETCI	GET-COLL-INPUT	FUNC	6/25		Tst	used by COLL-COORD.
SCNQUE	SCAN-QUEUE	FUNC	7/15		Code	used by MAC
NFYOPN	NOTIFY-OPNS	FUNC	7/15		Code	used by MAC
BRFASM	BRIEF-ASSESSMT	FUNC	7/15		Code	used by MAC

Table A-1 Translation of DHIL Statements (page 9 of 9)

APPENDIX B. ENGLISH LANGUAGE SCENARIOS

The following narratives are summaries of the scenarios presented in the project document "Conceptual I&W Data Base Specification," dated 15 July 1981, and later revised by Mc^2. They are intended to give the reader a general picture of the type of activity described in the scenarios, and should not be used as the basis for detailed implementation or study. (Appendix C contains a precise specification of the scenarios in the DHIL format.) Note that three of the nine scenarios listed here were not fully implemented (as indicated in the title lines). The first section (numbered B.1) contains a generalized scenario for I&W analysis and is not intended for implementation.

B.1 Generalized I&W Scenario

This scenario presents a generalized task sequence for the I&W analyst. Emphasis is placed on the structure of I&W analysis, rather than on specific actions or data requirements.

The analyst has been tasked to provide a specific analysis. This tasking may be either a continuing responsibility or a special request. To obtain appropriate data to support the analysis, the analyst must first define the bounds of time and location of the situation being analyzed.

Within these bounds, the analyst calls for and reviews military activities of a designated type (such as troop maneuvers). If no significant activities are found, then the analyst continues by searching for other activities of interest. (Note that "significant" must be defined for each type of analysis. An event is significant only within the context of the analyst's task assignment.)

While reviewing military events of several types within the designated space-time boundaries, the analyst finds a report that may be significant (such as an increased number of aircraft sighted at a particular location within the area of interest). At this point in time, the significance of the report has not been determined, and its correctness has not been verified. The next steps are intended to establish the significance and correctness of the report.

The analyst extends the search to include earlier and later time periods, reviewing military activities of various types as before. If nothing relevant or significant is found, the analyst eventually abandons this analysis and returns to the earlier mode, perhaps marking the report for future reference. The two modes might be called "seek" and "confirm." The first is a search for items that suggest a hypothesis of interest; the second is a search for items that confirm, deny, or expand upon the hypothesis. The first mode is comparatively undirected, somewhat like the lookout in the crow's nest scanning the horizon for items of interest; the second is a directed search for the required information, as the lookout attempts to verify and identify an obscure patch that appears in the telescope.

Another extension of the initial search includes adjacent geographical areas. The expanded search may also include related military or political activities, background information on the area, and any other relevant information. The analyst's search is intended to obtain information which confirms, explains, and expands upon the initial hypothesis. Additional data may be retrieved from the data base, data from one file may be compared with data from another file, data may be displayed and combined with other data against the background of a map on a color terminal, graphs and histograms may be drawn to help visualize the data, and various statistical routines may be employed to assist in the analysis.

At some point, which is defined by the type of confirmatory data obtained, the significance of the event, and the time requirements for warning or other reports, the analyst may prepare a briefing or draft document summarizing the findings of this search, including references to the sources of information and relevant quotations from them. A report generator and other editing facilities assist in this task. The analyst's immediate superior normally reviews and approves the report, which is then prepared and transmitted in the appropriate form. The analyst returns to a review of incoming message traffic.

B.2 Watch Function

Identify and determine significance of current and impending events which may require a U.S. military response, or which may affect the ability to carry out a military response.

The simulated Watch Officer is reviewing incoming messages within an assigned area of interest in order to identify and report on events which may require a military response from the U.S. In general, the messages are routine, but a small fraction will be regarded as unusual. (In the simulation, a random process determines whether to regard a message as interesting or unusual. The actual content of the message is not taken into account in the RTE simulation.)

When an unusual message is identified, the analyst studies it closely to determine whether further action is required. This judgment is based on the analyst's prior knowledge and general understanding of the area of concern. In most instances, there will be no need for further action, and the analyst will return to the incoming message file. (Again, a random process determines whether the simulated analyst will regard a message as significant.)

A message is received which indicates that a coup appears to be imminent within the analyst's area of interest. The analyst uses the system to retrieve other messages that deal with events occurring at about the same time and place, and which will serve to verify and expand upon the information contained in the first message. The analyst also examines the current list of indicators for this area to determine the extent and significance of the coup. A statistical algorithm is invoked to predict the time of occurrence of the coup, based on indicator values.

The next step for the analyst is to determine whether friendly missions will include flight routes which may be affected by the coup. A query to the data base obtains this information. Records concerning these flights are marked for subsequent analysis. The analyst now studies the related information that has been retrieved to determine whether further attention is warranted. If not, the analyst returns to the task of reviewing incoming messages. (The decision whether to regard the impending coup as significant is simulated by a random process.)

If the coup will have an impact on flights which are within the analyst's area of responsibility, a brief text is prepared and transmitted to the Route Assessment officer. The data base is updated to reflect the threat to U.S. flights and to U.S. nationals in the area. A notation is added to the data concerning this country indicating that there is an impending coup.

The messages used during this analysis are now returned to the file, and the analyst returns to the routine task of reviewing incoming messages. When a predetermined number of messages have been reviewed, the scenario terminates.

B.3 Military Systems Analysis

Identify significant changes in military system capabilities which may require a U.S. military response.

The analyst is reviewing messages which are concerned with foreign military system capabilities. Most of these are routine and report recent changes, which are entered in the data base to update current information. Sometimes a message indicates a change in system status which will require further analysis to establish its significance for friendly military forces and activities. (The occurrence of significant changes is simulated by a random process.)

Messages containing information on system status are selected by the analyst and studied to determine the need for confirmation and further review. If further study is warranted, the analyst enters appropriate instructions to locate additional information concerning the weapon system from the Equipment Summary. The range of the system is reviewed to determine the extent of the threat that it presents.

The simulated analyst now consults with other analysts, using the computer terminal to send messages to persons with some knowledge of the weapon system. From information gathered in this way, the analyst uses the terminal to build a more complete picture of the potential threat. A cartographic program builds a display of the geographic area involved, and the location referenced in the initial message is indicated. In this scenario, the equipment type is a SAM, and the range of this equipment is indicated on the map, which is displayed at the terminal. A computer algorithm is used to determine impact assessment by correlating the capabilities of the missile installation with preplanned friendly flight routes. The analyst studies this information to determine whether the installation will have a significant effect *upon those routes*. (The analyst's judgment is simulated by a random process.)

If there will be a significant impact on friendly flight routes, the analyst prepares briefing material, extracting necessary information from the data just retrieved. A warning message is prepared and disseminated. Flight routes are modified to take account of the new threat. The analyst then returns to the task of reviewing incoming messages. After a predetermined number of messages have been processed, the scenario terminates.

B.4 Collection Coordination

Coordinate intelligence collection requests with current tasking.

The collection coordinator is scanning incoming messages which contain requests for collection. These are correlated to bring together requests for the same type of information in the same geographic area. Available reconnaissance data are reviewed to determine whether the required information has already been made available, and to determine current reconnaissance tasking.

If current requests will not be met by current tasking, new requirements must be transmitted. Priorities are established for the new tasks and cleared with the analyst's division chief. A report generator is used to produce a collection request form, which is added to the list of collection requirements. The analyst returns to the earlier task of reviewing incoming requests. When a pre-set number of requests have been processed, the scenario terminates.

B.5 MAC Route Assessment

Review MAC current operations and routes, and identify threats.

This scenario represents the actions of an analyst responsible for reviewing MAC current operations and routes to identify potential threats posed by nations on or near the flight routes. These threats are identified through the use of information contained in a file of new SAM sites.

In this scenario, as the analyst is reviewing current operations schedules, it is noted that an embassy evacuation has been ordered. Since any evacuation presents a relatively high risk, the analyst will perform a detailed assessment of the route being flown to determine what air defense threats may affect the mission.

The first action of the analyst is to retrieve and review the route to be used in the evacuation. The information to support this portion of the analysis is obtained by retrieving and reviewing the data base file entries for each route segment identified in the initial message.

Next, the analyst reviews all the data retrieved to determine whether planned MAC missions are threatened. If the analyst determines that no threat exists, the scenario returns to the beginning; if there appears to be a threat, the analysis continues. (This decision is simulated by a random process.)

The next step is to begin building a composite display at the terminal. Cartographic data for the area of interest are used to create a map, and various overlays are constructed on the map. Among these are: aircraft routings, restricted areas, order of battle information, and terrorist data. The information used to construct these overlays is contained in the data retrieved by the initial queries to the data base.

Once maps and overlays have been constructed for the area of interest, the analyst will continue with the threat evaluation previously begun. If there appears to be a threat, the analyst uses the computer system to contact operations personnel concerned with the flight under examination to determine the degree to which they are aware of the threat.

After consulting with the operations staff, the analyst prepares a Route Threat Summary consisting chiefly of the information contained on the map and overlays developed earlier in the session. Next, the analyst prepares a Flight Following Log.

The analyst now monitors ongoing message traffic pertinent to the MAC mission being tracked to determine whether the original threat assessment should be changed. If there is an increased threat, the additional information is communicated to operations personnel. A warning message is prepared and transmitted.

The analyst then prepares a mission summary using information contained in the warning message. Once the data base has been updated, the analyst will return to the routine planning task with which the scenario began. When a pre-set number of executions have been completed, the scenario terminates.

B.6 I&W Analysis

The analyst is reviewing incoming message traffic, updating routine message files as required. An occasional I&W input message requires further analysis, and such a message is selected by the analyst from the input queue and stored in a temporary file. The analyst next turns to a file of current holdings for a review of other related events. If no related information is available, the search parameters are extended, and the search continues until there *is sufficient collateral information* to provide a context for the message.

After a careful study of this related information, the analyst uses graphics software to build a flight map for the area under study. The AOB is queried to obtain data from which flights over this area are plotted. An algorithm for impact assessment is invoked, identifying flight routes of friendly missions. These flight routes are plotted on the map displayed at the analyst's terminal. After further study, the analyst may decide that a warning will be required. (This decision is simulated in the scenario by a random process.)

The analyst discusses the situation with the collection coordinator to determine whether further information may be obtained. Next, the indicator list is reviewed to find out whether any relevant indicators have been triggered. On the basis of information already gathered, the analyst updates the indicator list. Next, the flight routes of friendly missions are reviewed to determine the times at which potential conflicts may occur. Using computer generated map displays, the analyst now presents a briefing on the apparent threat to friendly missions. Following approval from the division chief, the analyst prepares a warning message using editing facilities available at the terminal, and the warning is transmitted through on-line message dissemination facilities.

The scenario maintains a count of messages transmitted. When this count exceeds a pre-set number, the scenario terminates; otherwise, the simulated analyst returns to the task of reviewing input messages.

B.7 Area Analysis

The analyst is reviewing incoming messages at the terminal. Many of them are irrelevant or routine and require no more than routine treatment, using storage faci:ities available from the terminal. A message is received which indicates an increased alert status at an East German air field, and which appears to require a more extensive analysis. The analyst temporarily stores this new message.

The analyst now compares the input message with current message holdings. Specifically, the analyst enters a request for information concerning the status of the unit at the base which was identified in the message. The information consists primarily of counts of the number and status of various types of aircraft at the base. If there is no confirmation of the apparent alert, the analysis terminates, and the analyst returns to routine message review. (The decision to terminate the analysis is simulated by a random process.)

With confirmation, however, the analyst updates the file to indicate the increased alert status. If necessary, additional data will be obtained to provide further information concerning the unit. (Again, the decision to retrieve further information is simulated by a random process.) The analyst's next request is for a report of all surface-to-air missile activity within a given radius of the airfield under study. Cartographic capabilities at the computer terminal permit the analyst to build a map of the area, indicating SAM location. Threat areas around the SAM installations are plotted, based on equipment and terrain parameters. An overlay is prepared and displayed indicating the unit locations. The analyst also notes that SAM sites and units have been observed on increased alert.

The analyst next consults with the collection coordinator to obtain any further information, and produces an impact assessment. This includes an identification of 40 friendly missions which may be affected by the observed enemy activities. Friendly missions are plotted on the terminal display against the map background. The analyst now has created a visual display which permits a study of the effect of enemy activities on friendly missions. If a threat is indicated, a notation, "Watch for SAM," is entered into the data base.

If the situation warrants, the analyst next examines the status of current Soviet and East German indicators for the area under study. These confirm the report of increased alert at the first air base. To document this conclusion, hardcopy output is produced from the terminal display. The analyst updates the indicators to reflect the new activity. Next, the analyst prepares a briefing, based on the information collected. A warning message is prepared with the aid of editing facilities available at the terminal, and submitted to the Division Chief for review and approval. Following approval, the message is transmitted, and added to the current message files.

At this point, the required analysis has been completed, and the analyst returns to the task of examining incoming messages. After a pre-set number of messages have been examined, the scenario terminates.

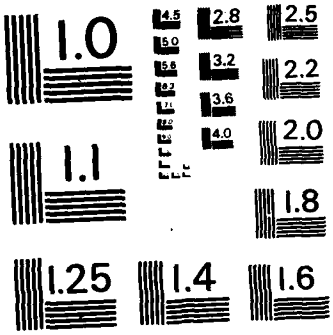
B.8 Space Event Assessment (not implemented)

Determine possible threat associated with space vehicle launch which may require U.S. military response.

The analyst is reviewing incoming message traffic. For this analyst, space vehicles launched from Site A are studied and reported. The initial task is to locate messages in which the Launch Site Name is Site A. When such a report is located, the information is saved in a temporary storage area, and the analyst requests additional data concerning activities at that site and nearby sites within a specified time span. These additional records are added to the information already saved.

AD-A139 339 I&W (INDICATIONS AND WARNING) DATA BASE MANAGEMENT 3/4
SYSTEMS ANALYSIS(U) MEASUREMENT CONCEPT CORP ROME NY
JUN 83 RADC-TR-83-136 F30602-80-C-0286

UNCLASSIFIED F/G 5/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Using the information gathered during this search, the analyst calls upon a support program to compute the initial orbit of the space vehicle. Next, information concerning the booster type of this vehicle is retrieved through the system and added to the summary records. The completed Launch Event Summary is produced by the system's report generator and transmitted to the analyst's immediate superior for review and approval.

The analyst now turns to other activities. The first act is to retrieve space vehicle trajectory information, based on the event identification code of this space event. A description of the satellite is plotted on the display terminal and a subprogram is called to determine the degree of threat. (The existence of a threat, for the purposes of the scenario, is determined by a random process.) If the space vehicle does represent a threat, a message is generated to report the existence of this threat. The message is submitted to the analyst's immediate superior for approval and dissemination.

Following preparation of the warning message, the analyst returns to the task of reviewing incoming messages. When the analyst has cycled through this process a predetermined number of times, the scenario ends.

B.9 Missile Threat Assessment (not implemented)

Determine possible threat associated with a missile launch which may require a U.S. military response.

The analyst is reviewing incoming messages, which have been pre-selected as potentially relevant to missile threat assessment. A message concerning a missile launch appears to be potentially relevant, and the next task will be to verify the reported event and to determine the threat, if any, that the launch presents.

Certain data are reviewed to determine whether the missile type is of interest. If not, then the event is recorded, and the analyst goes on to review the next record. If characteristic features of the missile indicate that the event is of interest, the analyst proceeds to review the event in greater detail.

The first step is to plot the location of the launch. A map of the area is displayed on the terminal, and the location is identified. The appropriate order of battle data are retrieved from the data base, identifying the missile through the reported data and other available information. The equipment status summary is also displayed, and the circle range is computed with the help of a computer program. Another program is called to compute the trajectory of the missile. Finally, information concerning friendly installations which may be threatened by the missile is also retrieved, displayed, and placed in temporary storage.

Next, the analyst reviews other information which will be used in determining the existence and extent of the threat. (In the actual execution of the scenario, a random process is used to decide whether the reported missile represents a threat.) If no threat exists, the analyst returns to reviewing message traffic. If the analyst determines that there is sufficient evidence to warrant an increased level of alert, this information is transmitted to the analyst's immediate superior for appropriate action. Next, an air defense alert briefing is prepared, based on retrieved information, using the automated display and report generation facilities of the system.

The current list of active indicators is modified to show the launch. A program is called to compute residual threat based on information retrieved from the OB. Other records in the data base are reviewed and updated to show the launch. For the purposes of the simulation, a count is kept of the number of times the scenario has looped through this sequence of activities. After a pre-set number of loops, the scenario terminates; otherwise, the simulated analyst returns to reviewing incoming messages.

B.10 ELINT Analysis (not implemented)

Determine possible threat associated with ELINT capabilities.

The analyst is reviewing ELINT reports within a designated area of responsibility. A continuing task for this analyst is updating the EOB, which is modified to reflect new information received. The analyst next calls on appropriate routines to plot the location and coverage of a specific radar installation. A map is displayed, and radar coverage is indicated, based on data concerning the radar type and terrain characteristics.

The analyst must next determine the threat posed by the installation. (A random process determines whether this analysis is actually required.) The ELINT Systems Summary is retrieved from the data base to aid in this analysis. Other analysts are consulted to obtain further information and evaluation. An impact assessment is developed, based on flight routes of friendly missions within the range of the radar under study. (In the scenario, a random process is used to decide whether a threat is present.)

If it appears to the analyst that the radar installation represents a threat to friendly missions, an initial assessment is presented to the analyst's immediate superior for approval. Using the report generation facilities of the DBMS, together with display facilities, the analyst prepares a briefing on the installation and prepares a warning message for dissemination. During execution of the scenario, a count is kept of the number of times this sequence of activities has been performed. After a predetermined number of executions, the scenario terminates.

B.11 Updates

In addition to the primary scenarios described above, an additional scenario provides a series of updates to the data base. The primary purpose of the scenario is to provide a flow of updates to simulate this portion of the load on the data base management system, to determine the system's ability to preserve data integrity in near real-time processing. For the purposes of the simulation, this scenario represents the clerical task of making new entries to the OB files, indicator lists, weather reports, and other portions of the data base.

Appendix C. SCENARIOS IN DHIL

Six scenarios, representing various types of analyst activities, based upon the following six intelligence tasks, were defined in DHIL and translated into scripts:

- o Watch
- o I&W Analysis
- o Collection Coordination
- o Area Analysis
- o Military Systems Analysis
- o MAC Route Assessment

This appendix contains complete DHIL versions of the scenarios, together with flowcharts indicating the sequence of analyst actions. The flowcharts are contained in Figures C-1 through C-6, with the accompanying DHIL versions following each flowchart.

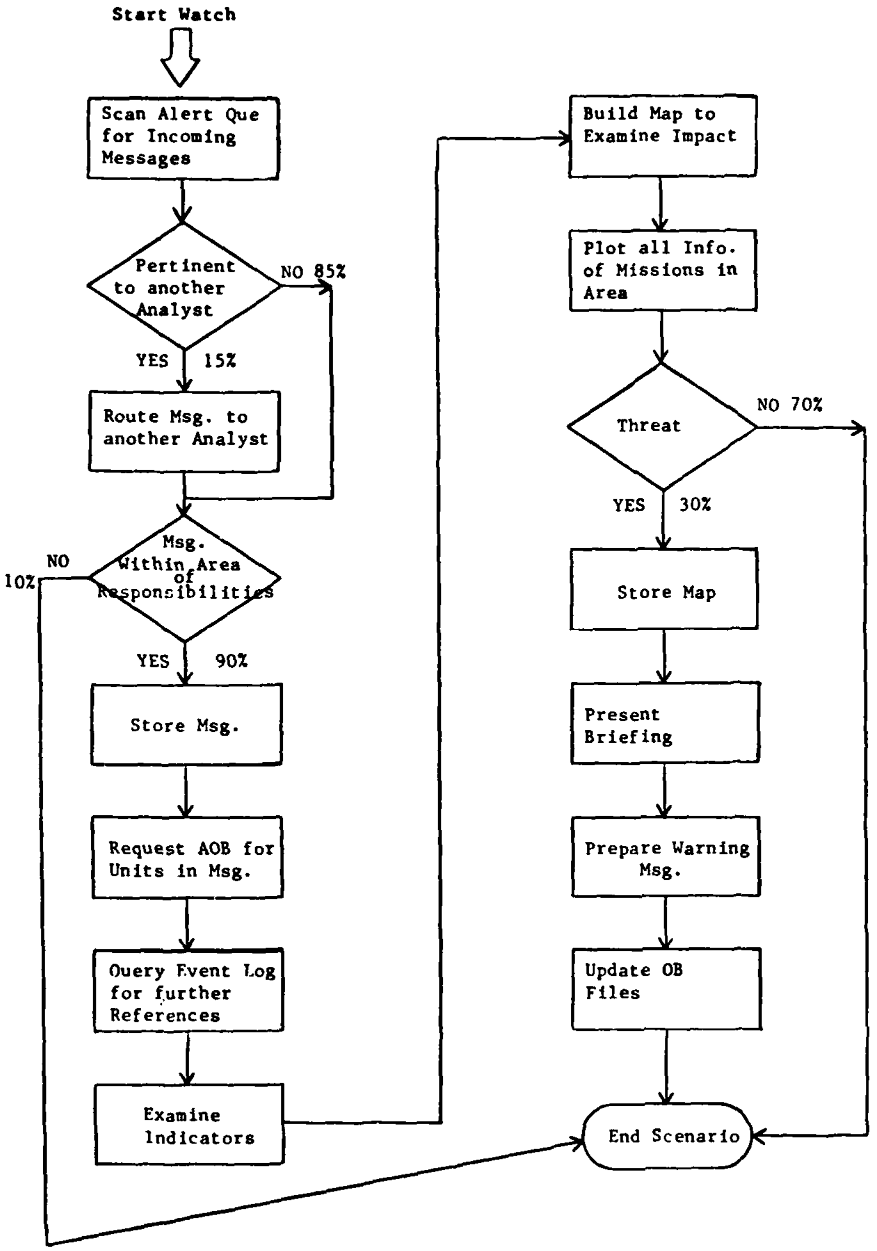

Start Watch
Scan Alert Que for Incoming Messages
Pertinent to another Analyst
NO 85%
YES
15%
Route Msg. to another Analyst
Msg. Within Area of Responsibilities
NO
10%
YES
90%
Store Msg.
Request AOB for Units in Msg.
Query Event Log for further References
Examine Indicators
Build Map to Examine Impact
Plot all Info. of Missions in Area
Threat
NO 70%
YES
30%
Store Map
Present Briefing
Prepare Warning Msg.
Update OB Files
End Scenario

Figure C-1 WATCH Function

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: WATCH-FUNCTION          $$WATCH.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-SCEN WATCH-FUNCTION;
C                 COMMENT:  THE WATCH OFFICER SCANS THE ALERT QUEUE
C                           TO REVIEW INCOMING MESSAGES;
C                 PERFORM REVIEW-INPUTS;
C                 COMMENT:  THE ANALYST ANALYZES THIS MESSAGE TO
C                           DETERMINE IF IT IS PERTINENT TO THE
C                           RESPONSIBILITIES OF ANOTHER ANALYST;
C                 SOMETIMES(.15,,);
C                      BEG-SOME;
C                      PERFORM ROUTE-MSG;
C                      END-SOME;
C                 COMMENT:  DECIDE IF THE MESSAGE IS WITHIN THE
C                           FUNCTIONAL AND GEOGRAPHIC AREA OF
C                           RESPONSIBILITY OF THE WATCH OFFICER,
C                           CAUSING THE INVESTIGATION TO CONTINUE;
C                 SOMETIMES(.90,,);
C                      BEG-SOME;
C                      PERFORM STORE-MSG;
C                      COMMENT:  THE ANALYST REQUESTS THE AIR ORDER
C                                OF BATTLE DATA FOR THE UNIT(S)
C                                IDENTIFIED IN THE INPUT MESSAGE;
C                      PERFORM COMP-MSG-HOLDINGS;
C                      COMMENT:  THE ANALYST DECIDES TO QUERY THE
C                                EVENT LOG FOR FURTHER REFERENCES;
C                      PERFORM REVIEW-EVENTS;
C                      COMMENT:  ANALYST RETRIEVES AND DISPLAYS
C                                INDICATOR LISTS FOR ASSESSMENT;
C                      PERFORM EXAMINE-SIGNF-INDICATORS;
C                      COMMENT:  ANALYST WILL BUILD MAP OF AREA OF
C                                CONCERN, OVERLAY FRIENDLY OPERATIONS
C                                DATA AND EXAMINE ANY POTENTIAL IMPACT
C                                BASED ON PREVIOUSLY RETRIEVED DATA;
C                      PERFORM BUILD-MAP;
C                      COMMENT:  ANALYST REQUESTS AND PLOTS ALL INFO.
C                                ON MISSIONS SCHEDULED IN THE AREA;
C                      PERFORM GET-CURRENT-OPNS;
C                      COMMENT:  THE ANALYST EVALUATES THIS AND PREVIOUS
C                                DATA TO DETERMINE IF A POSSIBLE THREAT
C                                EXISTS;
C                      SOMETIMES(.30,A,);
C                           BEG-SOME;
C                           COMMENT:  WHEN A THREAT MAY EXIST, THE
C                                     ANALYST PREPARES TO BRIEF THE
C                                     DIV-CHIEF/SUPERVISOR;
C                           PERFORM STORE-MAP;
C                           PERFORM BRIEF-TEXT;
C                           PERFORM PREP-WARNING-MSG;
C                           COMMENT:  THE ANALYST MUST NOW UPDATE THE
C                                     APPROPRIATE DB FILE(S);
C                           PERFORM UPDATE-FILES;
C                           END-SOME;
C                      END-SOME;
C            END-SCEN WATCH-FUNCTION.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: REVIEW-INPUTS            **REVINP.DRA**
C
C
C       ABSTRACT:
C
C            BEG-FUNC REVIEW-INPUTS;
C                 COMMENT: DISPLAY NUMBER AND KIND OF EACH ENTRY IN QUEUE;
C                 ALERT;
C                 THINK(10,30,A);
C                 COMMENT: DISPLAY SHORT TITLE - ALL QUEUED ALERTS;
C                 SCAN;
C                 THINK(10,30,A);
C                 COMMENT: SELECTIVELY DISPLAY AND READ A MESSAGE FROM
C                          THOSE ROUTED THROUGH THE ALERT QUEUE;
C                 RETRIEVE (A001.1, A001#1 EQ '$');
C                 THINK(30,90,A);
C                 PAGE;
C                 THINK(30,90,A);
C            END-FUNC  REVIEW-INPUTS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE-MSG                **RTEMSG.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC ROUTE-MSG;
C                 THINK(10,30,);
C                 COMMENT:  ROUTE THE APPROPRIATE MESSAGE INFORMATION
C                           TO THE RECEIVING ANALYST'S ALERT QUEUE;
C                 ROUTE(LOGIN = '$', MSG-ID = A001#1, PRI = A001#71,
C                       SUBJ = A001#190);
C                 THINK(5,15,);
C            END-FUNC ROUTE-MSG.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: STORE-MSG                 **STOMSG.FTN**
C
C
C      ABSTRACT:
C
C           BEG-FUNC STORE-MSG;
C                COMMENT:  STORE THE INPUT MESSAGE IN A TEMPORARY
C                          PERSONAL FILE FOR LATER USE;
C                STORE;
C           END-FUNC STORE-MSG.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: COMP-MSG-HOLDINGS         **CMPHLD.ORA**
C
CC
C      ABSTRACT:
C
C           BEG-FUNC COMP-MSG-HOLDINGS;
C                COMMENT:  THE ANALYST WILL REQUEST SPECIFIC INFO.
C                          ON UNITS (UP TO THREE) IDENTIFIED IN
C                          THE INPUT MESSAGE;
C                SOMETIMES(.50,,LOOP = 3);
C                     BEG-SOME;
C                     THINK(5,10,);
C                     COMMENT:  RETRIEVE AOB ON UNIT-NUMBER AND CLASS;
C                     RETRIEVE(C001.1,C001#1 EQ '$' AND C001#4 EQ 'FIS');
C                     COMMENT:  ANALYST READS THE INFORMATION;
C                     THINK(30,90,A);
C                     PAGE;
C                     THINK(30,90,A);
C                     END-SOME;
C           END-FUNC COMP-MSG-HOLDINGS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: REVIEW-EVENTS          ###RVEVNT.ORA###
C
C
C       ABSTRACT:
C
C       BEG-FUNC  REVIEW EVENTS;
C          COMMENT:  THE ANALYST REQUESTS ALL INFO. ABOUT EVENTS OF A
C               PARTICULAR TYPE, OCCURRING WITHIN THE COUNTRY IN QUESTION,
C               WHERE THE PARTICIPANTS SHOW AN ALLEGIANCE TO A PARTICULAR
C               COUNTRY (BASED ON INPUT MSG);
C          QUERY(B001.1, (B001#23 EQ C001#40 OR B001#23 EQ '$') AND
C               B001#11 NE '$' AND B001#32 EQ '$');
C          THINK(60,150,A);
C          PAGE;
C          THINK (60,150,A);
C          COMMENT:  THE ANALYST MAY REQUEST FURTHER INFORMATION ON
C               PERSONALITIES INVOLVED IN A PARTICULAR EVENT;
C          SOMETIMES (.20, , );
C             BEG-SOME;
C             SEARCH (D002.2, D002#200 EQ B001#2);
C                BEG-SEARCH;
C                THINK (2,5,);
C                COMMENT:  THE ANALYST DISPLAYS AND QUICKLY SCANS THE INFO;
C                REVIEW;
C                THINK (30,60,A);
C                PAGE;
C                THINK (30,60,A);
C                END-SEARCH;
C             END-SOME;
C       END-FUNC  REVIEW-EVENTS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: EXAMINE-SIGNF-INDICATORS  ##EXSIND.ORA##
C
C
C       ABSTRACT:
C               BEG-FUNC EXAMINE-SIGNIF-INDICATORS;
C                  COMMENT: THE ANALYST REQUESTS COUNTRY SUMMARY INFORMATION;
C                  RETRIEVE(E001.1,E001#2 EQ C001#40);
C                  COMMENT: THE ANALYST SCANS THIS INFORMATION,THEN REQUESTS
C                          ALL INDICATORS OF A CERTAIN TYPE FOR THIS COUNTRY;
C                  QUERY(F001.1,F001#1 EQ C001#40 AND F001#21 EQ '$');
C                  COMMENT: ANALYST SCANS INDICATOR LISTS;
C                  THINK(30,150,A);
C                  PAGE;
C                  THINK(30,150,A);
C               END-FUNC EXAMINE-SIGNF-INDICATORS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: BUILD-MAP                 **BUILDM.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC BUILD-MAP;
C                 COMMENT:  CREATE MAP BACKGROUND WITH UNIT
C                           PLACED AT THE CENTER OF THE MAP;
C                 MAP(LAT = C001#41, LON = C001#41A,);
C                 THINK(10,30,);
C            END-FUNC BUILD-MAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: GET-CURRENT-OPNS       ***GETOPS.ORA***
C
C
C       ABSTRACT:
C
C            BEG-FUNC GET-CURRENT-OPNS;
C                 COMMENT: REQUESTS ALL MISSIONS WHOSE DESTINATION IS THE
C                          COUNTRY IN QUESTION;
C                 SEARCH(H001.1,H001#7 EQ C001#40);
C                      BEG-SEARCH;
C                      THINK(2,5, );
C                      REVIEW;
C                      THINK(120,300,A);
C                      COMMENT: ANALYST READS GENERAL MISSION INFO. AND
C                               DECIDES IF FURTHER INVESTIGATON IS NECESSARY;
C                      SOMETIMES(.80,A, );
C                        BEG-SOME;
C                        COMMENT: ANALYST PLOTS THE DEPARTURE POINT AND TAGS
C                                 WITH THE MSN-ID;
C                        PLOT( ,CLASS = MSN);
C                        THINK(5,10, );
C                        INSERT( , ,BLOCK = H001#2);
C                        COMMENT: THE ANALYST DECIDES TO DIRECTLY PLOT ALL
C                                 END-POINTS FOR EACH MISSION SEGMENT;
C                        PLOT(I002.1,I002#70 EQ H001#2,CLASS = 'SEG',NOBLOCK);
C                        COMMENT: THE ANALYST CAN CONSTRUCT THE ENTIRE ROUTE
C                                 USING THE LIGHTPEN TO CONNECT THE SEG END-POINTS;
C                        THINK(5,10, );
C                      END-SOME;
C                 END-SEARCH;
C            END-FUNC GET-CURRENT-OPNS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: STORE-MAP               **STOMAP.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC STORE-MAP;
C                 COMMENT:  STORE THE FULLY CONSTRUCTED MAP FOR
C                           LATER USE IN BRIEFINGS;
C                 STORE;
C            END-FUNC STORE-MAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: BRIEF-TEXT              **BRFTXT.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC BRIEF-TEXT;
C                 COMMENT:  ANALYST DISPLAYS FULLY CONSTRUCTED
C                           MAP AND DISCUSSES THE ASSESSMENT;
C                 RECALL;
C                 THINK(60,125,A);
C                 COMMENT:  ANALYST DISPLAYS THE MESSAGE THAT
C                           LED TO THIS CONCLUSION;
C                 RECALL;
C                 THINK(60,125,A);
C            END-FUNC BRIEF-TEXT.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PREP-WARNING-MSG          $$PRWMSG.FTN$$
C
C
C       ABSTRACT:
C               BEG-FUNC PREP-WARNING-MSG;
C                 COMMENT: ANALYST DISPLAYS MESSAGE FORMAT AND ENTERS
C                          APPROPRIATE DATA;
C                 FORM A001.1;
C                 THINK(5,10, );
C                 ENTER(A001#1 = '$',A001#71 = 'I',
C                    A001#190 = 'UNUSUAL INDICATOR',
C                    A001#200 = 'MESSAGE TEXT -- DETAILS OF EVALUATION');
C                 THINK(30,60, );
C                 COMMENT: THIS WARNING MESSAGE WILL BE ADDED TO THE DATABASE
C                          MESSAGE FILE, THE ANALYST WILL MAKE A PERSONAL COPY,
C                          THEN ROUTE THE MESSAGE TO THE SUPERVISOR'S ALERT
C                          QUEUE;
C                 ADDON A001.1;
C                 HARDCOPY;
C                 THINK(5,10, );
C                 ROUTE(LOGIN = 'DIV-CHIEF',MSG-ID = A001#1,PRI = 'I',
C                      SUBJ = 'UNUSUAL INDICATOR');
C               END-FUNC PREP-WARNING-MSG.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: UPDATE-FILES             $$UPDFIL.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC UPDATE-FILES;
C                 THINK(30,60,A);
C                 COMMENT:  THE ANALYST DECIDES TO UPDATE THE COUNTRY
C                           SUMMARY FILE;
C                 RETRIEVE(E001.1,E001#2 EQ C001#40);
C                 THINK(10,20,);
C                 MODIFY(E001#15 = 'HIGH', E001#16 = 'IMPENDING COUP');
C            END-FUNC UPDATE-FILES;
r

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ALERT
C
C
C       ABSTRACT:
C
C          ALERT will simulate an analyst displaying numbers and types
C       of messages on the alert queue.  The actual messages are not relevant.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: THINK
C
C
C       ABSTRACT:
C
C          THINK simulates an analyst mulling over a decision.  Because of
C       this, the result is divided by the alert status.  Obviously, the
C       analyst will think less time in a crisis or war situation.
C
C
C       INPUT:          MIN     - INTEGER;  Minimum peacetime wait
C                       MAX     - INTEGER;  Maximum wait at any time
C                       ALERT   - INTEGER;  Alert status (1,2,3)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SCAN
C
C
C       ABSTRACT:
C
C          SCAN simulates an analyst scanning through the list of alert
C       messages.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE                   **ROUTE.FTN**
C       ABSTRACT:
C
C            The DHIL command ROUTE is used to insert message information
C       into the alert queue of an analyst.
C
C       The RTE command format is:
C       ROUTE(LOGIN = <receiving analyst>, MSG-ID = <field value>,
C             PRI = <field value>, SUBJ = <field value>))
C
C       Required parameters are:
C       LOGIN  -- the login identifier of the receiving analyst
C       MSG-ID -- the msg-id-code which uniquely identifies a message
C                 within the databas
C       PRI    -- the message priority
C       SUBJ   -- brief information indicating the subject/content of
C                 the message
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: MAP                         **MAP.FTN**
C
C
C       ABSTRACT:
C
C       THE DHIL 'MAP' COMMAND IS USED TO GENERATE A MAP BACKGROUND.
C
C       THE RTE COMMAND FORMAT IS:
C       MAP(COORD = <seo.coords.> ! LOC = <location name>,
C            [SCALE = <scale>]);
C
C       PARAMETERS ARE:
C       COORD!LOC - MUST SPECIFY EITHER COORD OR LOC (BUT NOT BOTH)
C                   TO IDENTIFY THE CENTER POINT OF THE DISPLAY
C
C       SCALE     - OPTIONAL PARAMETER WHICH SPECIFIES THE SCALE OF THE
C                   MAP TO BE DISPLAYED.  SCALES AVAILABLE ARE: 1:20M,
C                   1:5M, 1:1M, 1:500K, 1:250K, 1:100K, 1:50K, 1:25K,
C                   1:10K, 1:7.5K.  IF NOT SPECIFIED, DEFAULT TO 1:20M
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: INSERT                   **INSERT.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'INSERT' COMMAND IS USED TO ADD A NEW ITEM TO THE
C       SPECIFIED LOCATION ON THE CURRENT MAP DISPLAY WITH A DATA BLOCK
C       APPENDED TO IT, OR TO ASSOCIATE A DATA BLOCK WITH AN ITEM ALREADY
C       ON THE CURRENT MAP DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       INSERT([CLASS = <symbol>],[LOC = <value>],BLOCK = <value>);
C
C       PARAMETERS ARE:
C       CLASS = <symbol> - OPTIONAL PARAMETER WHICH SPECIFIES THE SYMBOLOGY
C                          CLASSIFICATION OF THE ITEM BEING ADDED TO THE
C                          DISPLAY.
C       LOC = <value>    - OPTIONAL PARAMETER WHICH SPECIFIES THE LOCATION
C                          VALUE OF THE ITEM BEING ADDED TO THE DISPLAY.
C                          IF NOT GIVEN, DEFAULT TO THE CURRENT LOCATION
C                          OF THE CURSOR/LIGHT PEN.
C       BLOCK = <value>  - REQUIRED PARAMETER WHICH SPECIFIES THE DATA BLOCK
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: STORE
C
C
C       ABSTRACT:
C
C            THE DHIL 'STORE' COMMAND IS USED TO SAVE THE CURRENT SCREEN
C       DISPLAY (MAP/RECORD) IN A TEMPORARY FILE FOR FAST RECALL AT A LATER
C       TIME.  ONLY THREE 'RECORDS' CAN BE STORED AT ANY TIME, AND THEY ARE
C       ACCESSED ON A LAST-IN FIRST-OUT BASIS VIA THE 'RECALL' COMMAND.
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: RECALL                   $$RECALL.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'RECALL' COMMAND IS USED TO ACCESS A MAP/RECORD
C       WHICH HAS BEEN STOREd IN A TEMPORARY FILE.
C
C       THE RTE WILL EXECUTE THIS COMMAND AS A 'WAIT' TIME
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: FORM                       $$FORM.FTN$$
C
C
C       ABSTRACT:
C
C            The FORM command is used to display the template
C       format for the specified userview.
C
C       The RTE command format is:
C            FORM <userview>;
C       This command is used to display a form which will then
C       have information ENTERed into it by the analyst (a sequence
C       of steps used to add a record to the database).  For testing
C       purposes this command will be executed as a WAIT time.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ENTER                         $$ENTER.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'ENTER' COMMAND IS USED TO SIMULATE THE ANALYST
C       ENTERING DATA INTO A PREVIOUSLY DISPLAYED FORM.  ONLY THE CONTENTS
C       OF THE DISPLAY ARE MODIFIED -- THIS COMMAND HAS NO EFFECT ON THE
C       DATABASE.  SUBSEQUENT PRIMITIVE DIRECTIVES CONTROL THE DISPOSITION
C       OF THE DATA.
C
C       THE RTE COMMAND FORMAT IS:
C       ENTER(<fieldname> = <new value>[,<fieldname> = <new value>]);
C
C       PARAMETERS ARE:
C            <fieldname> - THE NAME OF THE FIELD RECEIVING DATA
C            <new value> - VALUE TO BE ASSIGNED TO RECEIVING FIELD
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE                         $$ROUTE.FTN$$
C
C
C       ABSTRACT:
C
C            The DHIL command ROUTE is used to insert message information
C       into the alert queue of an analyst.
C
C       The RTE command format is:
C       ROUTE(LOGIN = <receiving analyst>, MSG-ID = <field value>,
C             PRI = <field value>, SUBJ = <field value>);
C
C       Required parameters are:
C       LOGIN  -- the login identifier of the receiving analyst
C       MSG-ID -- the msg-id-code which uniquely identifies a message
C                 within the databas
C       PRI    -- the message priority
C       SUBJ   -- brief information indicating the subject/content of
C                 the message
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: HARDCOPY                 $$HRDCPY.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'HARDCOPY' COMMAND IS USED TO FORMAT AND TRANSFER
C       ALL OF THE CURRENT DISPLAY TO A HARDCOPY DEVICE.
C
C       THE RTE COMMAND FORMAT IS:
C            HARDCOPY;
C       THE COMMAND WILL BE ISSUED AS A WAIT TIME FOR THE BENCHMARK
C       EXERCISES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CMPRS   - COMPRESS
C
C
C       ABSTRACT:
C
C          CMPRS will compress any buffer by removing all spaces, tabs, and
C       nulls.  The remainder of the line up to the specified number of
C       characters is padded with blanks.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

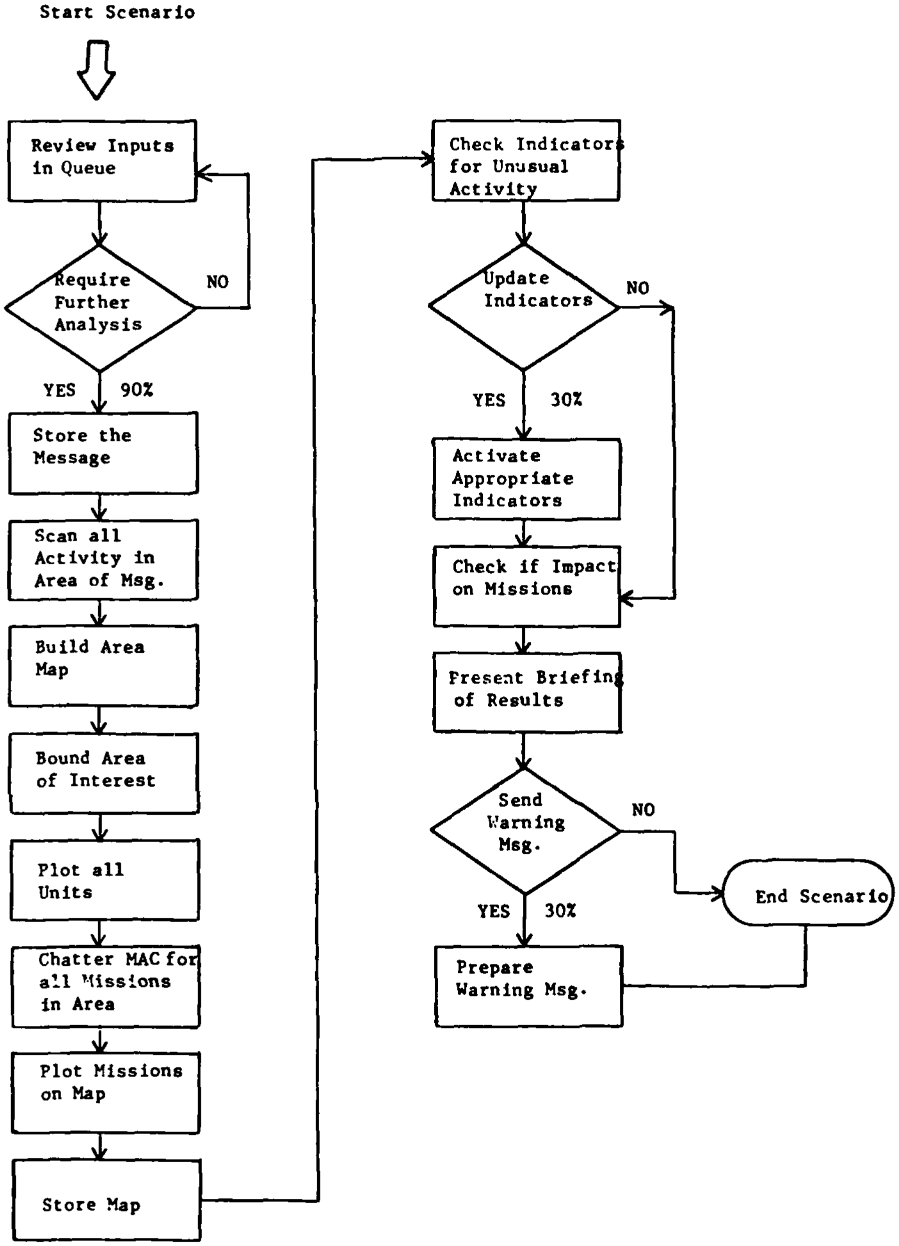


Figure C-2 I & W Analyst

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME:   IW ANALYSIS
C
C
C       ABSTRACT:
C               BEG-SCEN IW-ANALYSIS;
C                  COMMENT: ANALYSTS SCANS ALERT QUEUE FOR INCOMING MESSAGES;
C                  PERFORM REVIEW-INPUTS;
C                  COMMENT: ANALYST DECIDES IF THE MESSAGE RETRIEVED IS
C                           PERTINENT AND REQUIRFS FURTHER ANALYSIS;
C                  SOMETIMES(.90, , );
C                  BEG-SOME;
C                       PERFORM STORE-MSG;
C                       COMMENT: THE ANALYST BEGINS GY SCANNING THE DATABASE FOR
C                                INFO ON RECENT ACTIVITY WITHIN THE AREA OF INTEREDT;
C                       PERFORM FIND-ALL-ACTIVITY;
C                       COMMENT: ANALYST BUILDS A MAP OF THE AREA;
C                       PERFORM BUILD-AREA-MAP;
C                       COMMENT: IN ATTEMPTING TO ROUND THE AREA OF SFARCH THE
C                                ANALYST RETRIEVES INFO ON THE AIRCRAFT SYSTEM
C                                REFERRED TO IN THE MESSAGE;
C                       PERFORM BOUND-AREA;
C                       COMMENT: UNDER THE ASSMPTION THAT NO IN-AIR REFUELING
C                                HAS TAKEN PLACE,THE ANALYST WILL ATTEMPT TO
C                                LOCATE THE UNITS ASSOCIATED WITH THIS SYSTEM,
C                                AND PLOT THESE UNITS ON THE MAP;
C                       PERFORM PLOT-AOB;
C                       COMMENT: THE ANALYST QUERIES MAC FOR ALL MISSIONS
C                                SCHEDULED IN THIS AREA;
C                       PERFORM CHATTER-MAC;
C                       COMMENT: OVERLAY AIR ROUTE DATA ON MAP;
C                       PERFORM PLOT-FLT-ROUTES;
C                       PERFORM STORE-MAP;
C                       COMMENT: ANALYST RETRIEVES AND DISPLAYS INDICATOR LISTS;
C                       PERFORM EXAMINE-INDICATORS;
C                       COMMENT: THE ANALYST WILL ACTIVATE APPROPRIATE INDICATORES;
C                       SOMETIMES(.30,A, );
C                          BEG-SOME;
C                          PERFORM ACTIVATE-INDICATORS;
C                          END-SOME;
C                       COMMENT: ANALYST MUST DETERMINE IF (AND WHEN)CONFLICTS
C                                MIGHT EXIST FOR FRIENDLY FLIGHT ACTIVITY IN
C                                THE AREA UNDER SCRUNITY;
C                       PERFORM COMPUTE-CONFLICT-TIMES;
C                       THINK(30,60,A);
C                       COMMENT: ANALYST PREPARES BRIEFING ON ASSESSMENT IN ORDER
C                                TO GAIN CONCURRENCE AND/OR APPROVAL OF THE C
C                                ANALYTICAL RESULTS;
C                       PERFORM BRIEF-TEXT;
C                       COMMENT: FOLLOWING APPROVAL OF THE ASSESSMENT/BRIEFING
C                                ,THE ANALYST IS RESPONSIBLE FOR PREPARING A
C                                WARNING MESSAGE WHEN NECESSARY;
C                       SOMFTIMES(.30,,);
C                          BEG-SOME;
C                          PERFORM PREP-WARNING-MSG;
C                          END-SOME;
C               END-SCEN IW-ANALYSIS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C
C       PROCEDURE NAME: REVIEW-INPUTS           **REVINP.DRA**
C
C       ABSTRACT:
C
C            BEG-FUNC REVIEW-INPUTS;
C                 COMMENT: DISPLAY NUMBER AND KIND OF EACH ENTRY IN QUEUE;
C                 ALERT;
C                 THINK(10,30,A);
C                 COMMENT: DISPLAY SHORT TITLE - ALL QUEUED ALERTS;
C                 SCAN;
C                 THINK(10,30,A);
C                 COMMENT: SELECTIVELY DISPLAY AND READ A MESSAGE FROM
C                          THOSE ROUTED THROUGH THE ALERT QUEUE;
C                 RETRIEVE (A001.1, A001#1 EQ '$');
C                 THINK(30,90,A);
C                 PAGE;
C                 THINK(30,90,A);
C            END-FUNC  REVIEW-INPUTS.
C
C
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C
C       PROCEDURE NAME: STORE-MSG               **STOMSG.FTN**
C
C       ABSTRACT:
C
C            BEG-FUNC STORE-MSG;
C                 COMMENT:  STORE THE INPUT MESSAGE IN A TEMPORARY
C                           PERSONAL FILE FOR LATER USE;
C                 STORE;
C            END-FUNC STORE-MSG.
C
C
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: FIND-ALL-ACTIVITY        $$FNDACT.ORA$$
C
C
C      ABSTRACT:
C              BEG-FUNC FIND-ALL-ACTIVITY;
C                 COMMENT: QUERY THE EVENT FILE FOR RECENT ACTIVITY
C                          OCCURRING WITHIN THE SPECIFIED COUNTRY;
C                 QUERY(B001.1,B001#23 EQ '$' AND B001#11 EQ '$');
C                 COMMENT: ANALYST RETRIEVES AND SCANS EACH RECORD FOUND;
C                 THINK(30,300,A);
C                 PAGE;
C                 THINK(30,300,A);
C              END-FUNC FIND-ALL-ACTIVITY.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: BUILD-AREA-MAP  $$$BLDMAP.FTN$$$
C
C
C      ABSTRACT:
C              BEG-FUNC BUILD-AREA-MAP;
C                 COMMENT: ANALYST BUILDS A MAP OF THE AREA REFERRED TO IN
C                          THE MESSAGE;
C                 MAP(LAT = '$' ,LON = '$', );
C                 THINK(2,5, );
C              END-FUNC BUILD-AREA-MAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: BOUND-AREA               $$BNDARE.ORA$$
C
C
C      ABSTRACT:
C              BEG-FUNC BOUND-AREA;
C                   RETRIEVE(D001.2,D001#1 EQ 'FIGHTER' AND D001#2 EQ '$');
C                   THINK(10,20,A);
C              END-FUNC BOUND-AREA.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT-AOB                    **PLTAOB.ORA**
C
C
C       ABSTRACT:
C               BEG-FUNC PLOT-AOB;
C                  COMMENT: ANALYST SEARCHES FOR UNITS WITHIN THE MAX. RANGE
C                           OF THE SIGHTING OF THE AIRCRAFT WHICH ARE KNOWN
C                           TO HAVE THIS SYSTEM  TYPE;
C                  CSEARCH(C001.1,C001*111A EQ 'FIGHTER' AND
C                          C001*111B EQ D001*2,LAT = '$',LON = '$',
C                          RANGE = 1000KM), PERFORM NOT-IDENTIFIED;
C                       BEG-SEARCH;
C                       THINK(2,5, );
C                       COMMENT: ANALYST SCANS EACH UNIT RECORD FOUND AND PLOTS
C                                ONTO MAP;
C                       REVIEW;
C                       THINK(60,120,A);
C                       PAGE;
C                       THINK(30,60,A);
C                       PLOT( ,CLASS = C001*1, );
C                       END-SEARCH;
C               END-FUNC PLOT-AOB.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CHATTER-MAC     ***CHTMAC.FTN***
C
C
C       ABSTRACT:
C               BEG-FUNC CHATTER-MAC;
C                  INSERT(,,BLOCK = 'ANY MISSIONS IN THIS AREA?');
C                  THINK(5,10,);
C                  CHATTER MAC-ROUTE-THREAT-ASSESS;
C                  COMMENT: THE ANALYST WAITS FOR MAC TO RESPOND WITH
C                           MISSION-IDs FOR ALL OPERATIONS IN THE AREA;
C                  THINK(300,600,);
C                  SCAN;
C                  THINK(5,10,);
C                  DISPLAY;
C                  THINK(20,30,);
C               END-FUNC CHATTER-MAC.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: PLOT-FLT-ROUTES          **PLTRTE.FTN**
C
C
C      ABSTRACT:
C              BEG-FUNC PLOT-FLT-ROUTES;
C                 COMMENT: THE ANALYST WILL REQUEST ALL MISSION INFORMATION
C                          TO OVERLAY ON PREVIOUSLY CONSTRUCTED MAP,
C                          BASED ON THE INFO. MAC SENT;
C                 SOMETIMES(.30,A,LOOP = 3);
C                      BEG-SOME;
C                      RETRIEVE(H001.1,H001#2 EQ '#');
C                      THINK(120,300,A);
C                      COMMENT: PLOT THE DEPARTURE POINT AND TAG EACH ROUTE
C                               WITH ITS MISSION-ID;
C                      PLOT(,CLASS = 'MSN',);
C                      THINK(5,10, );
C                      INSERT( , ,BLOCK = H001#2);
C                      COMMENT: THE ANALYST REQUESTS AND PLOTS ALL
C                               SEGMENT INFORMATION;
C                      SEARCH(I002.1,I002#70 EQ H001#2);
C                         BEG-SEARCH;
C                         THINK(5,10, );
C                         REVIEW;
C                         THINK(10,30, );
C                         SEG( , ,TOP LAT = I002#7,TOP LON = I002#7A)
C                         END-SEARCH;
C                      END-SOME;
C          END-FUNC PLOT-FLT-ROUTES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: EXAMINE-INDICATORS       **EXMIND.ORA**
C
C      ABSTRACT:
C              BEG-FUNC EXAMINE-INDICATORS;
C                 COMMENT: THE ANALYST REQUESTS ALL MILITARY INDICATORS
C                          FOR THE COUNTRY OBSERVED;
C                 QUERY(F001.1,F001#1 EQ C001#40 AND F001#21 EQ 'MIL');
C                 THINK(10,15,);
C                 PAGE;
C                 THINK(10,15,);
C              END-FUNC EXAMINE-INDICATORS.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: ACTIVATE-INDICATORS      **ACTIND.ORA**
C
C
C      ABSTRACT:
C
C           BEG-FUNC ACTIVATE-INDICATORS;
C                COMMENT: UPDATE THE INDICATOR STATUS;
C                MODIFY(F001#24 = 'A');
C                COMMENT: THE ANALYST MAKES HARDCOPY FOR LATER REFERENCES;
C                HARDCOPY;
C                THINK(5,10,);
C              END-FUNC ACTIVATE-INDICATORS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: COMPUTE-CONFLICT-TIME   ***CMPTIN.FTN***
C
C      ABSTRACT:
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: BRIEF-TEXT                **BRFTXT.FTN**
C
C
C      ABSTRACT:
C
C           BEG-FUNC BRIEF-TEXT;
C                COMMENT:  ANALYST DISPLAYS FULLY CONSTRUCTED
C                          MAP AND DISCUSSES THE ASSESSMENT;
C                RECALL;
C                THINK(60,125,A);
C                COMMENT:  ANALYST DISPLAYS THE MESSAGE THAT
C                          LED TO THIS CONCLUSION;
C                RECALL;
C                THINK(60,125,A);
C           END-FUNC BRIEF-TEXT.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PREP-WARNING-MSG         ##PRWMSG.FTN##
C
C
C       ABSTRACT:
C               BEG-FUNC PREP-WARNING-MSG;
C                 COMMENT: ANALYST DISPLAYS MESSAGE FORMAT AND ENTERS
C                          APPROPRIATE DATA;
C                 FORM A001.1;
C                 THINK(5,10, );
C                 ENTER(A001#1 = '$',A001#71 = 'I',
C                    A001#190 = 'UNUSUAL INDICATOR',
C                    A001#200 = 'MESSAGE TEXT -- DETAILS OF EVALUATION');
C                 THINK(30,60, );
C                 COMMENT: THIS WARNING MESSAGE WILL BE ADDED TO THE DATABASE
C                          MESSAGE FILE, THE ANALYST WILL MAKE A PERSONAL COPY,
C                          THEN ROUTE THE MESSAGE TO THE SUPERVISOR'S ALERT
C                          QUEUE;
C                 ADDON A001.1;
C                 HARDCOPY;
C                 THINK(5,10, );
C                 ROUTE(LOGIN = 'DIV-CHIEF',MSG-ID = A001#1,PRI = 'I',
C                       SUBJ = 'UNUSUAL INDICATOR');
C               END-FUNC PREP-WARNING-MSG.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ALERT
C
C
C       ABSTRACT:
C
C          ALERT will simulate an analyst displaying numbers and types
C       of messages on the alert queue.  The actual messages are not relevant.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: THINK
C
C
C       ABSTRACT:
C
C          THINK simulates an analyst mulling over a decision.  Because of
C       this, the result is divided by the alert status.  Obviously, the
C       analyst will think less time in a crisis or war situation.
C
C
C       INPUT:          MIN     - INTEGER;  Minimum peacetime wait
C                       MAX     - INTEGER;  Maximum wait at any time
C                       ALERT   - INTEGER;  Alert status (1,2,3)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SCAN
C
C
C       ABSTRACT:
C
C          SCAN simulates an analyst scanning through the list of alert
C       messages.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: STORE
C
C
C       ABSTRACT:
C
C            THE DHIL 'STORE' COMMAND IS USED TO SAVE THE CURRENT SCREEN
C       DISPLAY (MAP/RECORD) IN A TEMPORARY FILE FOR FAST RECALL AT A LATER
C       TIME.  ONLY THREE 'RECORDS' CAN BE STORED AT ANY TIME, AND THEY ARE
C       ACCESSED ON A LAST-IN FIRST-OUT BASIS VIA THE 'RECALL' COMMAND.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: PLOT                        **PLOT.FTN**
C
C
C      THE DHIL 'PLOT' COMMAND IS USED TO DISPLAY DBMS ITEMS ONTO A
C      PREVIOUSLY GENERATED MAP BACKGROUND.
C
C      THE RTE COMMAND FORMAT IS:
C      PLOT([<userview>,<fieldname> <relop> <fieldvalue> [<losop>
C           <fieldname> <relop> <fieldvalue>]],CLASS = <symbol>,
C           [BLOCK|NOBLOCK]);
C
C      PARAMETERS ARE:
C      <userview>,<fieldname>...<fieldvalue> - OPTIONAL SELECTION LIST
C             WHICH WHEN SPECIFIED CAUSES A DATABASE ACCESS FOR ALL RECORDS
C             THAT MEET THE CRITERIA TO BE PLOTTED ONTO THE MAP.
C             IF NOT SPECIFIED, PLOT THE INFO. ASSOCIATED WITH THE RECORD
C             CURRENTLY BEING DISPLAYED.
C      CLASS - REQUIRED PARAMETER WHICH INDICATES THE CLASSIFICATION OF
C             SYMBOLIC DATA TO BE PRESENTED ON THE MAP
C      BLOCK|NOBLOCK - OPTIONAL PARAMETER WHICH SPECIFIES WHETHER A 'DATA
C             BLOCK/TAG' IS TO BE APPENDED TO EACH SYMBOL GENERATED FOR
C             DISPLAY.  BLOCK IS THE DEFAULT.
C
C      NOTE: THIS IS THE ONLY DHIL COMMAND THAT WILL EITHER BE EXECUTED AS
C           A 'WAIT' TIME (WHEN ISSUED WITH NO SELECTION LIST), OR MAKE A
C           DATABASE ACCESS (WHEN ISSUED WITH A SELECTION LIST).
C
C      **************************************************************
C      **************************************************************
C      ****THIS 'PLOT' FUNCTION SHOULD ONLY BE CALLED BY FUNCTIONS WHICH****
C      ****ARE ISSUING A DHIL PLOT COMMAND WITHOUT A SELECTION LIST********
C      **************************************************************
C      **************************************************************
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: NOT-IDENTIFIED  ***NOTID.FTN***
C
C
C      ABSTRACT:
C             BEG-FUNC NOT-IDENTIFIED;
C                COMMENT: THE ANALYST CONNOT IDENTIFY THE SOURCE OF THIS
C                        AIRCRAFT BASED ON CURRENT INFO., SO DECIDES TO
C                        NOTIFY THE DIV-CHIEF/SUPERVISOR.
C                RECALL;
C                THINK(5,10, );
C                ROUTE(LOGIN = 'DIV-CHIEF',MSG-ID = A00101,PRI = 'I',
C                     SUBJ = 'INSUFICIENT INFO. TO ID - PLEASE ADVISE' );
C             END-FUNC NOT-IDENTIFIED.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: INSERT                   $$INSERT.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'INSERT' COMMAND IS USED TO ADD A NEW ITEM TO THE
C       SPECIFIED LOCATION ON THE CURRENT MAP DISPLAY WITH A DATA BLOCK
C       APPENDED TO IT, OR TO ASSOCIATE A DATA BLOCK WITH AN ITEM ALREADY
C       ON THE CURRENT MAP DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       INSERT([CLASS = <symbol>],[LOC = <value>],BLOCK = <value>);
C
C       PARAMETERS ARE:
C       CLASS = <symbol> - OPTIONAL PARAMETER WHICH SPECIFIES THE SYMBOLOGY
C                          CLASSIFICATION OF THE ITEM BEING ADDED TO THE
C                          DISPLAY.
C       LOC = <value>    - OPTIONAL PARAMETER WHICH SPECIFIES THE LOCATION
C                          VALUE OF THE ITEM BEING ADDED TO THE DISPLAY.
C                          IF NOT GIVEN, DEFAULT TO THE CURRENT LOCATION
C                          OF THE CURSOR/LIGHT PEN.
C       BLOCK = <value>  - REQUIRED PARAMETER WHICH SPECIFIES THE DATA BLOCK
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CHATTER                  $$CHTTER.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'CHATTER' COMMAND IS USED TO SEND A COPY OF THE
C       CURRENT TERMINAL DISPLAY TO ANOTHER ANALYST THROUGH THE ALERT
C       QUEUE.
C
C       THE RTE COMMAND FORMAT IS:
C       CHATTER <receiving LOGIN identifier> [,<receiving LOGIN identifier>];
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: DISPLAY                  $$DISPLY.FTN$$
C
C
C       ABSTRACT:
C
C            THE DMIL 'DISPLAY' COMMAND IS USED TO DISPLAY AN ALERT
C       QUEUE ENTRY TO THE TERMINAL.   THE RTE FORMAT IS:
C
C       DISPLAY [<alert number>][INTO FORM <userview>]]
C
C       BOTH ARE OPTIONAL PARAMETERS, WHERE:
C       <alert number> - THE NUMBER OF THE ALERT QUEUE ENTRY TO BE
C                        DISPLAYED.  IF NOT SPECIFIED, DEFAULT TO
C                        HIGHEST PRIORITY ALERT QUEUE ENTRY.
C       INTO FORM <userview> - PARAMETER USED BY THE UPDATE SCENARIO
C                        TO DISPLAY AN ALERT QUEUE ENTRY INTO THE
C                        SPECIFIED <userview> FORMAT.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CMPRS   - COMPRESS
C
C
C       ABSTRACT:
C
C          CMPRS will compress any buffer by removing all spaces, tabs, and
C       nulls.  The remainder of the line up to the specified number of
C       characters is padded with blanks.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: EVEN
C
C
C       ABSTRACT:
C
C          EVEN will add one to an odd integer, and ignore an even integer.
C       This function is necessary for the ADABAS-M search buffer length,
C       which must be an even number.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SEG                          $$SEG.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'SEG' COMMAND IS USED TO DRAW A LINE SEGMENT
C       BETWEEN TWO POINTS ON A DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       SEG([FROM-LAT = <from-latitude>, FROM-LON = <from-longitude>],
C           TO-LAT = <to-latitude>, TO-LON = <to-longitude>);
C
C       PARAMETERS ARE:
C            FROM-LAT AND FROM-LON ARE OPTIONAL PARAMETERS WHICH SPECIFY
C       THE STARTING POINT OF THE SEGMENT TO BE DRAWN. IF NOT SPECIFIED,
C       DEFAULT TO THE CURRENT POSITION OF THE CURSOR/LIGHT PEN.
C            TO-LAT AND TO-LON ARE REQUIRED PARAMETERS WHICH SPECIFY THE
C       ENDING POINT OF THE SEGMENT TO BE DRAWN.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: HARDCOPY                  $$HRDCPY.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'HARDCOPY' COMMAND IS USED TO FORMAT AND TRANSFER
C       ALL OF THE CURRENT DISPLAY TO A HARDCOPY DEVICE.
C
C       THE RTE COMMAND FORMAT IS:
C            HARDCOPY;
C       THE COMMAND WILL BE ISSUED AS A WAIT TIME FOR THE BENCHMARK
C       EXERCISES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: RECALL                    $$RECALL.FTN$$
C
C       ABSTRACT:
C
C            THE DHIL 'RECALL' COMMAND IS USED TO ACCESS A MAP/RECORD
C       WHICH HAS BEEN STOREd IN A TEMPORARY FILE.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: FORM                          **FORM.FTN**
C
C
C       ABSTRACT:
C
C            The FORM command is used to display the template
C       format for the specified userview.
C
C       The RTE command format is:
C            FORM <userview>;
C       This command is used to display a form which will then
C       have information ENTERed into it by the analyst (a sequence
C       of steps used to add a record to the database).  For testing
C       purposes this command will be executed as a WAIT time.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ENTER                        **ENTER.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'ENTER' COMMAND IS USED TO SIMULATE THE ANALYST
C       ENTERING DATA INTO A PREVIOUSLY DISPLAYED FORM.  ONLY THE CONTENTS
C       OF THE DISPLAY ARE MODIFIED -- THIS COMMAND HAS NO EFFECT ON THE
C       DATABASE.  SUBSEQUENT PRIMITIVE DIRECTIVES CONTROL THE DISPOSITION
C       OF THE DATA.
C
C       THE RTE COMMAND FORMAT IS:
C       ENTER(<fieldname> = <new value>[,<fieldname> = <new value>]);
C
C       PARAMETERS ARE:
C            <fieldname> - THE NAME OF THE FIELD RECEIVING DATA
C            <new value> - VALUE TO BE ASSIGNED TO RECEIVING FIELD
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: HARDCOPY                   $$HRDCPY.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'HARDCOPY' COMMAND IS USED TO FORMAT AND TRANSFER
C       ALL OF THE CURRENT DISPLAY TO A HARDCOPY DEVICE.
C
C       THE RTE COMMAND FORMAT IS:
C            HARDCOPY;
C       THE COMMAND WILL BE ISSUED AS A WAIT TIME FOR THE BENCHMARK
C       EXERCISES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE                      $$ROUTE.FTN$$
C
C
C       ABSTRACT:
C
C            The DHIL command ROUTE is used to insert message information
C       into the alert queue of an analyst.
C
C       The RTE command format is:
C       ROUTE(LOGIN = <receiving analyst>, MSG-ID = <field value>,
C             PRI = <field value>, SUBJ = <field value>);
C
C       Required parameters are:
C       LOGIN  -- the login identifier of the receiving analyst
C       MSG-ID -- the msg-id-code which uniquely identifies a message
C                 within the databas
C       PRI    -- the message priority
C       SUBJ   -- brief information indicating the subject/content of
C                 the message
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

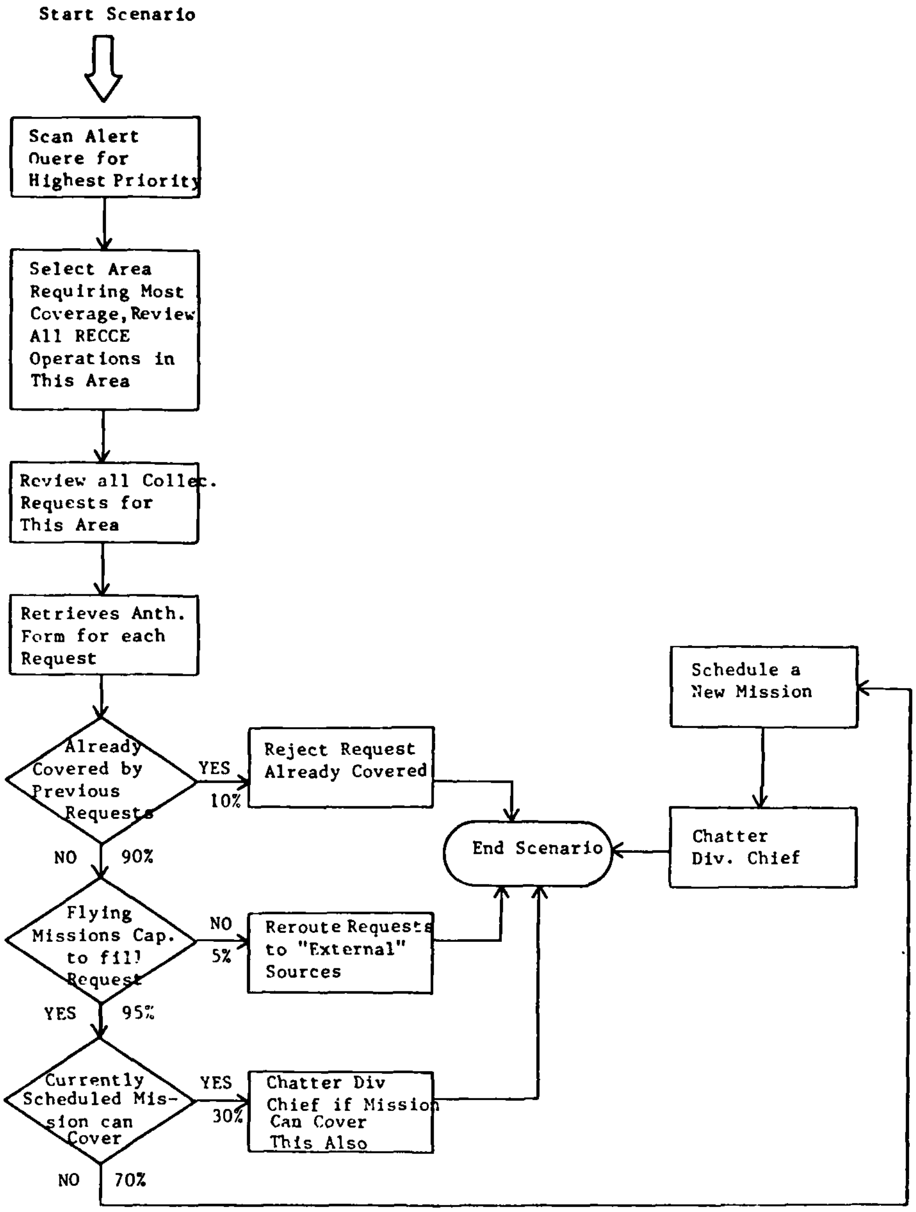

Start Scenario
Scan Alert Ouere for Highest Priority
Select Area Requiring Most Coverage, Review All RECCE Operations in This Area
Review all Collect. Requests for This Area
Retrieves Anth. Form for each Request
Already Covered by Previous Requests
YES
10%
Reject Request Already Covered
NO
90%
Flying Missions Cap. to fill Request
NO
5%
Reroute Requests to "External" Sources
YES
95%
Currently Scheduled Mission can Cover
YES
30%
Chatter Div Chief if Mission Can Cover This Also
NO
70%
End Scenario
Schedule a New Mission
Chatter Div. Chief

Figure C-3 Collection Coordination

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: COLLECTION-COORDINATION        ***COLCOR.DRA***
C
C
C      ABSTRACT:
C
C             BEG-SCEN COLLECTION-COORDINATION;
C                COMMENT: SCAN ALERT QUEUE TO SEE WHAT'S NEEDED;
C                PERFORM GET-COLL-INPUT;
C                COMMENT: THE ANALYST WILL SEARCH THE DATABASE FOR ALL
C                         HIGHEST PRIORITY REQUESTS, THE ANALYST LOOKS
C                         FOR ANY CORRELATIONS WHICH MIGHT ALLOW THE
C                         REQUESTS TO BE PROCESSED MORE EFFICIENTLY;
C                COMMENT: THE ANALYST PROCEEDS BY SEARCHING THE DATABASE
C                         FOR ALL REQUESTS THAT FIT THE 'CORRELATION',
C                         THEN PROCESSES THESE INDIVIDUALLY;
C                PERFORM EXTRACT-AND-PROCESS-REQUIREMENTS;
C             END-SCEN COLLECTION-COORDINATION.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: GET-COLL-INPUT           **GETCI.FTN**
C
C
C      ABSTRACT:
C
C           BEG-FUNC GET-COLL-INPUT;
C                COMMENT:  GET SOME DETAIL ON WHAT'S IN THE ALERT QUEUE;
C                ALERT;
C                THINK(2,5,);
C                SCAN;
C                THINK(10,30,A);
C                COMMENT:  DISPLAY HIGHEST PRIORITY ALERT;
C                DISPLAY;
C                THINK(10,30,A);
C           END-FUNC GET-COLL-INPUT.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C
C
C       ABSTRACT:
C
C               BEG-FUNC EXTRACT-AND-PROCESS-REQUIREMENTS;
C                  COMMENT: GET ALL HIGHEST PRIORITY REQUESTS;
C                  QUERY(G001.1,G001#6 EQ '$');
C                  THINK(10,60,);
C                  PAGE;
C                  THINK(10,60,);
•                  COMMENT: THE ANALYST WILL THEN SELECT THE AREA REQUIRING
C                           THE MOST COVERAGE AND REVIEW ALL RECCE OPERATIONS
C                           SCHEDULED FOR THIS AREA TO DETERMINE THE RESOURCES
C                           IMMEDIATELY AVAILABLE;
C                  PERFORM REVIEW-RECCE;
C                  COMMENT: THE ANALYST WILL THEN REVIEW ALL COLLECTION
C                           REQUESTS FOR THIS AREA;
C                  SEARCH(G001.1,G001#43 EQ J001#1);
C                       BEG-SEARCH;
C                       THINK(2,5,);
C                       COMMENT: THE ANALYST WILL EXAMINE AND PROCESS EACH
C                                REQUEST;
C                       REVIEW;
C                       THINK(30,60,);
C                       COMMENT: THE ANALYST RETRIEVES THE AUTHORIZATION
C                                FORM FOR THIS REQUEST;
C                       PERFORM GET-COLL-AUTHORIZATION;
C                       COMMENT: SOME REQUIREMENTS MAY ALREADY BE COVERED
C                                BY PREVIOUS REQUESTS;
C                       SOMETIMES(.10,,);
C                          BEG-SOME;
C                          PERFORM REJECT-REQUEST;
C                          END-SOME;
C                       COMMENT: WHEN A REQUEST WILL BE FILLED, THE COLLECTION
C                                COORDINATOR WILL DETERMINE WHETHER FLYING
C                                MISSIONS ARE CAPABLE OF FILLING THE REQUEST.
C                                IF 'EXTERNAL' SOURCES MUST BE CALLED UPON,
C                                THE ANALYST MUST ALERT THE SUPERVISOR;
C                       SOMETIMES(.5,,);
C                          BEG-SOME;
C                          PERFORM REROUTE-REQUEST;
C                          END-SOME;
C                       COMMENT: ONE OF THE CURRENTLY SCHEDULED RECCE OPERATIONS
C                                MAY BE CAPABLE OF HANDLING THIS REQUEST;
C                       SOMETIMES(.30,,);
C                          BEG-SOME;
C                          MODIFY(G001#111 = '$',G001#126 = 'CAN THIS MISSION
C                                     COVER THIS REQUEST,TOO?');
C                          THINK(5,10,);
C                          CHATTER DIV-CHIEF;
C                          END-SOME;
C                       COMMENT: SOME REQUESTS WILL REQUIRE THE SCHEDULING OF
C                                A NEW MISSION;
C                       SOMETIMES(.55,,);
C                          BEG-SOME;
C                          MODIFY(G001#126 = 'CAN THIS BE SCHEDULED?');
C                          THINK(5,10,);
C                          CHATTER DIV-CHIEF;
C                          END-SOME;
C                       END-SEARCH;
C                END-FUNC EXTRACT-AND-PROCESS-REQUIREMENTS.
C
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: REVIEW-RECCE             $$REVRCE.ORA$$
C
C
C       ABSTRACT:
C            BEG-FUNC REVIEW-RECCE;
C                 COMMENT: ANALYST REVIEWS AREA WEATHER;
C                 QUERY(J001.1,J001$1 EQ '$');
C                 THINK(10,30,);
C                 PAGE;
C                 THINK(10,30,);
C                 COMMENT: ANALYST REQUESTS INFO. ON ALL RECCE MISSIONS
C                          SCHEDULED IN THE AREA;
C                 SEARCH(H001.1,H001$1 EQ 'REC' AND H001$7 EQ J001$1);
C                      BEG-SEARCH;
C                      THINK(2,5,);
C                      REVIEW;
C                      THINK(120,300,A);
C                      END-SEARCH
C            END-FUNC REVIEW-RECCE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: GET-COLL-AUTHORIZATION    $$GETCA.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC GET-COLL-AUTHORIZATION;
C                 COMMENT:  THE ANALYST WILL FILL OUT ALL APPROPRIATE
C                           INFO. ON THIS FORM WHENEVER A COLLECTION
C                           REQUEST IS PROCESSED (DISAPPROVED/
C                           APPROVED, REASON/SCHEDULING INFO., ETC.);
C                 RETRIEVE(G001.2, G001$3 EQ G001$3);
C                 THINK(2,5,);
C            END-FUNC GET-COLL-AUTHORIZATION.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: REJECT-REQUEST           $$REJECT.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC REJECT-REQUEST;
C                 COMMENT:  NOTIFY REQUESTING ANALYST THAT THE INFORMATION
C                           DESIRED HAS ALREADY BEEN REQUESTED OR FILLED;
C                 MODIFY(G001$125 = 'D', G001$126 = 'YOUR INFO. IS ON THE
C                      WAY');
C                 THINK(5,10,);
C                 CHATTER ANALYST;
C            END-FUNC REJECT-REQUEST.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: REROUTE-REQUEST         ##REROUT.DRA##
C
C
C      ABSTRACT:
C
C           BEG-FUNC REROUTE-REQUEST;
C                MODIFY(G001#126 = 'CANNOT HANDLE THIS -- REROUTE');
C                THINK(5,10,);
C                CHATTER DIV-CHIEF;
C           END-FUNC REROUTE-REQUEST.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: ALERT
C
C
C      ABSTRACT:
C
C         ALERT will simulate an analyst displaying numbers and types
C      of messages on the alert queue.  The actual messages are not relevant.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: THINK
C
C
C      ABSTRACT:
C
C         THINK simulates an analyst mulling over a decision.  Because of
C      this, the result is divided by the alert status.  Obviously, the
C      analyst will think less time in a crisis or war situation.
C
C
C      INPUT:          MIN    - INTEGER;  Minimum peacetime wait
C                      MAX    - INTEGER;  Maximum wait at any time
C                      ALERT  - INTEGER;  Alert status (1,2,3)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: SCAN
C
C
C      ABSTRACT:
C
C         SCAN simulates an analyst scanning through the list of alert
C      messages.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: DISPLAY                 $$DISPLY.FTN$$
C
C
C      ABSTRACT:
C
C           THE DMIL 'DISPLAY' COMMAND IS USED TO DISPLAY AN ALERT
C      QUEUE ENTRY TO THE TERMINAL.   THE RTE FORMAT IS:
C
C      DISPLAY [<alert number>][INTO FORM <userview>]$
C
C      BOTH ARE OPTIONAL PARAMETERS, WHERE:
C      <alert number> - THE NUMBER OF THE ALERT QUEUE ENTRY TO BE
C                       DISPLAYED.  IF NOT SPECIFIED, DEFAULT TO
C                       HIGHEST PRIORITY ALERT QUEUE ENTRY.
C      INTO FORM <userview> - PARAMETER USED BY THE UPDATE SCENARIO
C                       TO DISPLAY AN ALERT QUEUE ENTRY INTO THE
C                       SPECIFIED <userview> FORMAT.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: CHATTER                 $$CHTTER.FTN$$
C
C
C      ABSTRACT:
C
C           THE DMIL 'CHATTER' COMMAND IS USED TO SEND A COPY OF THE
C      CURRENT TERMINAL DISPLAY TO ANOTHER ANALYST THROUGH THE ALERT
C      QUEUE.
C
C      THE RTE COMMAND FORMAT IS:
C      CHATTER <receiving LOGIN identifier> [,<receiving LOGIN identifier>]$
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: COLLECTION-COORDINATION        ***COLDEL.ORA***
C                       **U/DELETE**
C
C       ABSTRACT:
C
C               BEG-SCEN COLLECTION-COORDINATION;
C                  COMMENT: SCAN ALERT QUEUE TO SEE WHAT'S NEEDED;
C                  PERFORM GET-COLL-INPUT;
C                  COMMENT: THE ANALYST WILL SEARCH THE DATABASE FOR ALL
C                           HIGHEST PRIORITY REQUESTS, THE ANALYST LOOKS
C                           FOR ANY CORRELATIONS WHICH MIGHT ALLOW THE
C                           REQUESTS TO BE PROCESSED MORE EFFICIENTLY;
C                  COMMENT: THE ANALYST PROCEEDS BY SEARCHING THE DATABASE
C                           FOR ALL REQUESTS THAT FIT THE 'CORRELATION',
C                           THEN PROCESSES THESE INDIVIDUALLY;
C                  PERFORM EXTRACT-PROCESS-AND-DELETE-REQUIREMENTS;
C               END-SCEN COLLECTION-COORDINATION.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: EXTRACT-PROCESS-AND-DELETE-REQUIREMENTS
C                       $$EXTDEL.ORA$$
C
C
C       ABSTRACT:
C
C               BEG-FUNC EXTRACT-PROCESS-AND-DELETE-REQUIREMENTS;
C                  COMMENT: GET ALL HIGHEST PRIORITY REQUESTS;
C                  QUERY(G001.1,G001#6 EQ '$');
C                  THINK(10,60,);
C                  PAGE;
C                  THINK(10,60,);
C                  COMMENT: THE ANALYST WILL THEN SELECT THE AREA REQUIRING
C                           THE MOST COVERAGE AND REVIEW ALL RECCE OPERATIONS
C                           SCHEDULED FOR THIS AREA TO DETERMINE THE RESOURCES
C                           IMMEDIATELY AVAILABLE;
C                  PERFORM REVIEW-RECCE;
C                  COMMENT: THE ANALYST WILL THEN REVIEW ALL COLLECTION
C                           REQUESTS FOR THIS AREA;
C                  SEARCH(G001.1,G001#43 EQ J001#1);
C                       BEG-SEARCH;
C                       THINK(2,5,);
C                       COMMENT: THE ANALYST WILL EXAMINE AND PROCESS EACH
C                                REQUEST;
C                       REVIEW;
C                       THINK(30,60,);
C                       COMMENT: THE ANALYST RETRIEVES THE AUTHORIZATION
C                                FORM FOR THIS REQUEST;
C                       PERFORM GET-COLL-AUTHORIZATION;
C                       COMMENT: SOME REQUIREMENTS MAY ALREADY BE COVERED
C                                BY PREVIOUS REQUESTS;
C                       SOMETIMES(.10,,);
C                          BEG-SOME;
C                          PERFORM REJECT-REQUEST;
C                          END-SOME;
C                       COMMENT: WHEN A REQUEST WILL BE FILLED, THE COLLECTION
C                                COORDINATOR WILL DETERMINE WHETHER FLYING
C                                MISSIONS ARE CAPABLE OF FILLING THE REQUEST.
C                                IF 'EXTERNAL' SOURCES MUST BE CALLED UPON,
C                                THE ANALYST MUST ALERT THE SUPERVISOR;
C                       SOMETIMES(.5,,);
C                          BEG-SOME;
C                          PERFORM REROUTE-REQUEST;
C                          END-SOME;
C                       COMMENT: ONE OF THE CURRENTLY SCHEDULED RECCE OPERATIONS
C                                MAY BE CAPABLE OF HANDLING THIS REQUEST;
C                       SOMETIMES(.30,,);
C                          BEG-SOME;
C                          MODIFY(G001#111 = '$',G001#126 = 'CAN THIS MISSION
C                                    COVER THIS REQUEST,TOO?');
C                          THINK(5,10,);
C                          CHATTER DIV-CHIEF;
C                          END-SOME;
C                       COMMENT: SOME REQUESTS WILL REQUIRE THE SCHEDULING OF
C                                A NEW MISSION;
C                       SOMETIMES(.55,,);
C                          BEG-SOME;
C                          MODIFY(G001#126 = 'CAN THIS BE SCHEDULED?');
C                          THINK(5,10,);
C                          CHATTER DIV-CHIEF;
C                       DELETE;
C                       END-SEARCH;
C               END-FUNC EXTRACT-AND-PROCESS-REQUIREMENTS.
```

```
C       PROCEDURE NAME: ADD-TO-THE-G-FILE        **ADDTOG.FTN**
C
C
C       ABSTRACT:
C
C          NO FORMAL SCENARIO HAS BEEN WRITTEN FOR THIS PROCEDURE --
C       IT'S PURPOSE IS TO CONTINUALLY ADD NEW RECORDS TO THE G1A AND
C       G1AA FILE SO THAT THE VERSION OF THE COLLECTION-COORDINATION
C       SCENARIO WHICH DELETES RECORDS WILL NOT RUN OUT OF DATA.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: UPDATE                    **UPDATE.FTN**
C
C
C       ABSTRACT:
C
C            BEG-SCEN UPDATE:
C                 COMMENT:  MESSAGES ARE ENTERING THE QUEUE OF THE UPDATE
C                           SCENARIO AND WILL BE PROCESSED SEQUENTIALLY
C                           BASED ON PRIORITY OF MESSAGE;
C                 PERFORM MSG-INTO-FORM;
C                 COMMENT:  THE MESSAGE WILL HAVE A UNIQUE ID CODE
C                           ASSIGNED TO IT BEFORE ADDING IT TO THE
C                           DATABASE;
C                 PERFORM ADD-MSG-TO-DB;
C                 COMMENT:  DETERMINE WHO SHOULD SEE THIS MESSAGE AND
C                           ROUTE APPROPRIATE INFO. TO THEIR ALERT
C                           QUEUE.  IF RECEIVING ANALYST CANNOT BE
C                           DETERMINED, ROUTE TO WATCH OFFICER;
C                 SOMETIMES(.85,,);
C                      BEG-SOME;
C                      PERFORM ROUTE-TO-ANALYST;
C                      END-SOME;
C                 COMMENT:  WHEN IN DOUBT, ROUTE THE MESSAGE TO THE
C                           WATCH OFFICER;
C                 SOMETIMES(.20,,);
C                      BEG-SOME;
C                      PERFORM ROUTE-TO-WATCH;
C                      END-SOME;
C                 COMMENT:  WHEN THE MESSAGE IS FULLY PROCESSED IT
C                           SHOULD BE REMOVED FROM THE UPDATE QUEUE;
C                 PERFORM REMOVE-MESSAGE;
C            END-SCEN UPDATE.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: MSG-INTO-FORM             **MSGIF.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC MSG-INTO-FORM;
C                COMMENT:  OVERLAY INCOMING MESSAGE ONTO MSG. FORM;
C                DISPLAY INTO FORM A001.1;
C                THINK(2,5,);
C            END-FUNC MSG-INTO-FORM.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ADD-MSG-TO-DB            **ADDMSG.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC ADD-MSG-TO-DB;
C                 COMMENT:  ASSIGN ID CODE -- A COMPOUND FIELD WHICH
C                           INDICATES THE SOURCE AND DATE OF THE MESSAGE,
C                           AND A NUMBER INDICATING THE COUNT OF MESSAGES
C                           PROCESSED;
C                 ENTER(A001#1 = '$');
C                 THINK(2,5,);
C                 COMMENT:  THE MESSAGE CAN BE STORED IN THE DATABASE;
C                 ADDON A001.1;
C            END-FUNC ADD-MSG-TO-DB.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE-TO-ANALYST         **RANLST.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC ROUTE-TO-ANALYST;
C                 COMMENT:  UPDATE THE RECEIVING ANALYST'S QUEUE WITH
C                           THE MESSAGE-ID-CODE, THE MESSAGE PRIORITY,
C                           AND THE SUBJECT LINE OF THE MESSAGE;
C                 THINK(5,15,);
C                 ROUTE(LOGIN = '$', MSG-ID = A001#1, PRI = A001#71,
C                      SUBJ = A001#190);
C            END-FUNC ROUTE-TO-ANALYST.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE-TO-WATCH           **RWATCH.FTN**
C
C
C       ABSTRACT:
C
C            BEG-FUNC ROUTE-TO-WATCH;
C                 THINK(5,15,);
C                 ROUTE(LOGIN = 'WATCH-OFFICER', MSG-ID = A001#A,
C                      PRI = A001#71, SUBJ = A001#190);
C            END-FUNC ROUTE-TO-WATCH.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: DELETE                  &&DELETE.FTN&&
C
C
C       ABSTRACT:
C
C            The DELETE routine is used in conjunction with the
C       UPDATE scenario to prevent the ADABAS message files from
C       overflowing their maximum record limit.
C            When called, DELETE will simply remove all records from
C       the message files that have been created by the UPDATE scenario.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: DISPLAY                 &&DISPLY.FTN&&
C
C
C       ABSTRACT:
C
C            THE DHIL "DISPLAY" COMMAND IS USED TO DISPLAY AN ALERT
C       QUEUE ENTRY TO THE TERMINAL.   THE RTE FORMAT IS:
C
C       DISPLAY [<alert number>][INTO FORM <userview>]}
C
C       BOTH ARE OPTIONAL PARAMETERS, WHERE:
C       <alert number> - THE NUMBER OF THE ALERT QUEUE ENTRY TO BE
C                        DISPLAYED.  IF NOT SPECIFIED, DEFAULT TO
C                        HIGHEST PRIORITY ALERT QUEUE ENTRY.
C       INTO FORM <userview> - PARAMETER USED BY THE UPDATE SCENARIO
C                        TO DISPLAY AN ALERT QUEUE ENTRY INTO THE
C                        SPECIFIED <userview> FORMAT.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: THINK
C
C
C       ABSTRACT:
C
C          THINK simulates an analyst mulling over a decision.  Because of
C       this, the result is divided by the alert status.  Obviously, the
C       analyst will think less time in a crisis or war situation.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ENTER                     $$ENTER.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'ENTER' COMMAND IS USED TO SIMULATE THE ANALYST
C       ENTERING DATA INTO A PREVIOUSLY DISPLAYED FORM.  ONLY THE CONTENTS
C       OF THE DISPLAY ARE MODIFIED -- THIS COMMAND HAS NO EFFECT ON THE
C       DATABASE.  SUBSEQUENT PRIMITIVE DIRECTIVES CONTROL THE DISPOSITION
C       OF THE DATA.
C
C       THE RTE COMMAND FORMAT IS:
C       ENTER(<fieldname> = <new value>[,<fieldname> = <new value>]);
C
C       PARAMETERS ARE:
C            <fieldname> - THE NAME OF THE FIELD RECEIVING DATA
C            <new value> - VALUE TO BE ASSIGNED TO RECEIVING FIELD
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE                     $$ROUTE.FTN$$
C
C
C       ABSTRACT:
C
C            The DHIL command ROUTE is used to insert message information
C       into the alert queue of an analyst.
C
C       The RTE command format is:
C       ROUTE(LOGIN = <receiving analyst>, MSG-ID = <field value>,
C             PRI = <field value>, SUBJ = <field value>);
C
C       Required parameters are:
C       LOGIN  -- the login identifier of the receiving analyst
C       MSG-ID -- the msg-id-code which uniquely identifies a message
C                 within the databas
C       PRI    -- the message priority
C       SUBJ   -- brief information indicating the subject/content of
C                 the message
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

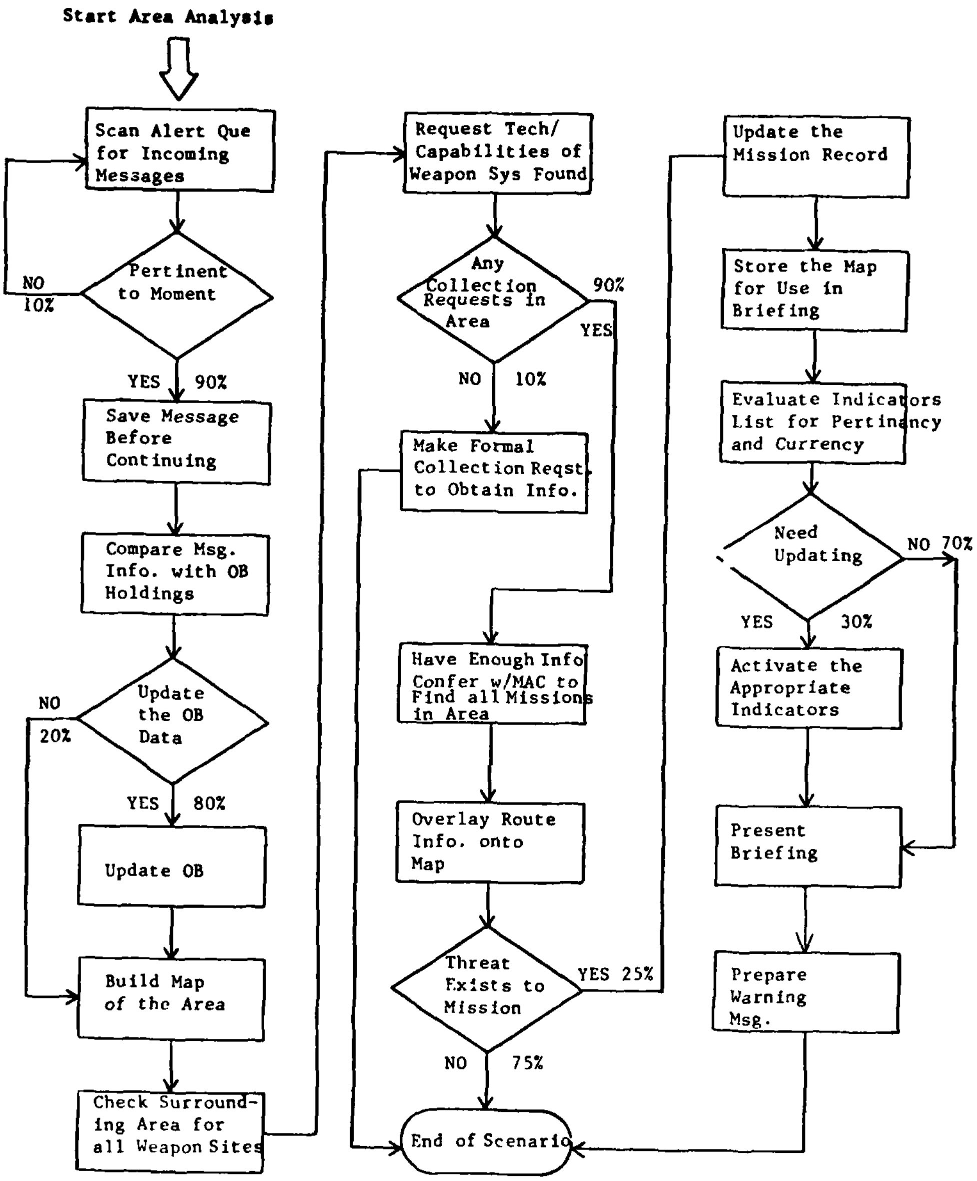

Start Area Analysis
Scan Alert Que for Incoming Messages
Pertinent to Moment
NO 10%
YES 90%
Save Message Before Continuing
Compare Msg. Info. with OB Holdings
Update the OB Data
NO 20%
YES 80%
Update OB
Build Map of the Area
Check Surrounding Area for all Weapon Sites
Request Tech/ Capabilities of Weapon Sys Found
Any Collection Requests in Area
90% YES
NO 10%
Make Formal Collection Reqst. to Obtain Info.
Have Enough Info Confer w/MAC to Find all Missions in Area
Overlay Route Info. onto Map
Threat Exists to Mission
YES 25%
NO 75%
End of Scenario
Update the Mission Record
Store the Map for Use in Briefing
Evaluate Indicators List for Pertinency and Currency
Need Updating
NO 70%
YES 30%
Activate the Appropriate Indicators
Present Briefing
Prepare Warning Msg.

Figure C-4 Area Analysis

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: AREA-ANALYSIS               $$AREA.ORA$$
C
C
C       AUTHOR:         , Mc2
C
C       ABSTRACT:
C
C            BEG-SCEN AREA-ANALYSIS;
C                 COMMENT: ANALYST SCANS ALERT QUEUE FOR INCOMING MESSAGES;
C                 PERFORM REVIEW-INPUTS;
C                 COMMENT: ANALYST DECIDES IF THE MESSAGE RETRIEVED IS
C                          PERTINENT AT THE MOMENT, AND SHOULD BE INVESTIGATED;
C                 SOMETIMES(.95,,);
C                       BEG-SOME;
C                       COMMENT: SAVE MESSAGE BEFORE CONTINUING;
C                       PERFORM STORE-MSG;
C                       COMMENT: ANALYST NOW COMPARES MSG INFO WITH
C                                OB HOLDINGS;
C                       PERFORM COMP-HOLDINGS;
C                       COMMENT: BASED ON THIS INFO. THE ANALYST MAY NEED TO
C                                UPDATE THE OB DATA;
C                       SOMETIMES(.80,A,);
C                            BEG-SOME;
C                            PERFORM UPDATE-OB;
C                            END-SOME;
C                       COMMENT: ANALYST BEGINS THE INVESTIGATION BY BUILDING A
C                                MAP OF THE AREA AROUND THIS UNIT;
C                       PERFORM BUILD-MAP;
C                       COMMENT: THE ANALYST DECIDES TO EVALUATE THE
C                                SURROUNDING AREA FOR ALL WEAPONS SITES;
C                       PERFORM ADDL-DATA-REQUIRED;
C                       THINK(15,30,A);
C                       COMMENT: REQUEST TECHNICAL CHARACTERISTICS/CAPABILITIES
C                                OF WEAPON SYSTEMS FOUND;
C                       PERFORM GET-CAPABILITIES;
C                       COMMENT: THE ANALYST DECIDES TO SEE IF ANY COLLECTIONS
C                                HAVE BEEN MADE IN THIS AREA RECENTLY;
C                       PERFORM CHATTER-COLL-COORD;
C                       COMMENT: IN SOME CASES, THE ANALYST MAY FIND IT
C                                NECESSARY TO MAKE A FORMAL COLLECTION REQUEST
C                                FOR INFO. NEEDED TO PROCEED WITH ASSESSMENT;
C                       SOMETIMES(.10,A,);
C                            BEG-SOME;
C                            PERFORM COLL-REQ;
C                            PERFORM DONE-PROCESSING;
C                            END-SOME;
C                       COMMENT: THE ANALYST HAS ENOUGH INFO. TO CONTINUE,
C                                AND SO DECIDES TO CONFER WITH MAC TO FIND
C                                ALL MISSIONS SCHEDULED IN THE AREA;
C                       PERFORM CHATTER-MAC;
C                       COMMENT: THE ANALYST CONTINUES ASSESSMENT BY OVERLAYING
C                                ROUTE INFO. ONTO MAP;
C                       PERFROM PLOT-FLT-ROUTES;
C                       COMMENT: THE ANALYST NOW MAKES AN ASSESSMENT OF THE
C                                POSSIBLE IMPACT ON CURRENT OPERATIONS;
C                       THINK(5,10,);
C                       COMMENT: WHEN THE ANALYST DETERMINES THAT A DEFINITE
C                                THREAT EXISTS, A BRIEFING IS PREPARED FOR THE
C                                DIVISION CHIEF/SUPERVISOR;
```

```
C                       SOMETIMES(.25,A,);
C                           BEG-SOME;
C                           COMMENT: THE ANALYST FIRST UPDATES THE MISSIO
C                                    RECORD;
C                           PERFORM MISSION-UPDATE;
C                           COMMENT: ANALYST SAVES A COPY OF THE MAP FOR
C                                    USE IN BRIEFING;
C                           PERFORM STORE-MAP;
C                           COMMENT: THE ANALYST REQUESTS INDICATOR LISTS
C                                    EVALUATE FOR PERTINENCE AND CURRENCY;
C                           PERFORM EXAMINE-INDICATORS;
C                           COMMENT: ANALYST WILL ACTIVATE APPROPRIATE
C                                    INDICATORS WHERE NECESSARY;
C                           SOMETIMES(.30,A,);
C                                BEG-SOME;
C                                PERFORM ACTIVATE-INDICATORS;
C                                END-SOME;
C                           PERFORM PRESENT-BRIEFING;
C                           PERFORM PREPARE-MSG;
C                       END-SOME;
C                  END-SOME;
C            END-SCEN AREA-ANALYSIS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: REVIEW-INPUTS           **REVINP.DRA**
C
C
C       ABSTRACT:
C
C            BEG-FUNC REVIEW-INPUTS;
C                 COMMENT: DISPLAY NUMBER AND KIND OF EACH ENTRY IN QUEUE;
C                 ALERT;
C                 THINK(10,30,A);
C                 COMMENT: DISPLAY SHORT TITLE - ALL QUEUED ALERTS;
C                 SCAN;
C                 THINK(10,30,A);
C                 COMMENT: SELECTIVELY DISPLAY AND READ A MESSAGE FROM
C                          THOSE ROUTED THROUGH THE ALERT QUEUE;
C                 RETRIEVE (A001.1, A001#1 EQ '$');
C                 THINK(30,90,A);
C                 PAGE;
C                 THINK(30,90,A);
C            END-FUNC  REVIEW-INPUTS.
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: STORE-MSG              **STOMSG.FTN**
C
C
C      ABSTRACT:
C
C          BEG-FUNC STORE-MSG;
C              COMMENT:  STORE THE INPUT MESSAGE IN A TEMPORARY
C                        PERSONAL FILE FOR LATER USE;
C              STORE;
C          END-FUNC STORE-MSG.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: COMP-HOLDINGS            **CMPHLD.ORA**
C
C
C      ABSTRACT:
C
C          BEG-FUNC COMP-HOLDINGS;
C              COMMENT: A REQUEST IS MADE FOR INFORMATION ON A
C                       SPECIFIC UNIT,IDENTIFIED IN THE INPUT MESSAGE;
C              RETRIEVE(C001.1,C001#1 EQ '$' AND C001#4 EQ '$');
C              THINK(60,120,A);
C              PAGE;
C              THINK(60,120,A);
C          END-FUNC COMP-HOLDINGS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: UPDATE-DB              **UPDTOB.ORA**
C
C
C      ABSTRACT:
C
C          BEG-FUNC UPDATE-DB;
C              MODIFY(C001#81 = 'ON ALERT');
C              THINK(5,10,);
C          END-FUNC UPDATE-DB.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: BUILD-AREA-MAP  $$$BLDMAP.FTN$$$
C
C
C       ABSTRACT:
C               BES-FUNC BUILD-AREA-MAP;
C                  COMMENT: ANALYST BUILDS A MAP OF THE AREA REFERRED TO IN
C                           THE MESSAGE; -
C                  MAP(LAT = '$' ,LON = '$', );
C                  THINK(2,5, );
C               END-FUNC BUILD-AREA-MAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC




CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ADDL-DATA-REQUIRED          $$ADLDAT.DRA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC ADDL-DATA-REQUIRED;
C                 COMMENT: A REQUEST IS MADE FOR INFO ON ALL SURFACE-TO-AIR
C                          MISSILES AND ANTI-AIRCRAFT ARTILLERY WITHIN A 200KM
C                          RANGE OF THE PREVIOUSLY IDENTIFIED UNIT;
C                 THINK(5,10,);
C                 CSEARCH(C001.1,C001#111A EQ 'SA' OR C001#111A EQ 'AA',
C                        LAT = C001#41,LON = C001#41A,RANGE = 200KM);
C                      BEG-SEARCH;
C                      THINK(2,5,);
C                      REVIEW;
C                      THINK(10,15,A);
C                      PAGE;
C                      THINK(10,15,A);
C                      COMMENT: ANALYST WILL SELECTIVELY PLOT THOSE UNITS
C                               THAT MAY HAVE AN IMPACT IN THIS AREA;
C                      SOMETIMES(.50,A,);
C                           BEG-SOME;
C                           PLOT( ,CLASS = C001#111A, );
C                           END-SOME;
C                      END-SEARCH;
C            END-FUNC ADDL-DATA-REQUIRED.
C
C
C
C       INPUT
C                      LDA    - logon data area
C                      LAT    - the latitude of the unit identified in the msg.
C                      LON    - the longitude of the same unit
C                      ALTFLG - alert status
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: GET-CAPABILITIES        **GETCAP.ORA**
C
C
C      ABSTRACT:
C
C           BEG-FUNC GET-CAPABILITIES;
C                COMMENT: RETRIEVALS MAY BE MADE ON UP TO 4 DIFFERENT SYSTEMS,
C                         BY CLASS AND TYPE;
C                SOMETIMES(.95,A,LOOP = 4);
C                      BEG-SOME;
C                      RETRIEVE(D001.1,D001#1 EQ '$' AND D001#2 EQ '$');
C                      THINK(15,30,);
C                      COMMENT: THE ANALYST INDICATES THE RANGE OF EACH
C                               SYSTEM ON THE MAP;
C                      CIRCLE(,RANGE = D001#40);
C                      THINK(5,10,);
C                      END-SOME;
C              END-FUNC GET-CAPABILITIES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: CHATTER-COLL-COORD      ***CHTCOL.FTN***
C
C
C      ABSTRACT:
C              BEG-FUNC CHATTER-COLL-COORD;
C                 INSERT(, ,BLOCK = 'HAVE ANYTHING ON THIS AREA ?');
C                 THINK(5,10, );
C                 CHATTER COLLECTION-COORDINATOR;
C                 COMMENT: AFTER SENDING A COPY OF THE MAP TO THE COLLECTION
C                          COORDINATOR, THE ANALYST WILL WAIT FOR A REPLY
C                          BEFORE CONTINUING;
C                 THINK(300,600, );
C                 COMMENT: THE ANALYST IS SCANNING THE ALERT QUEUE FOR A RESPONSE;
C                 SCAN;
C                 THINK(5,10, );
C                 DISPLAY;
C                 THINK(20,30, );
C              END-FUNC CHATTER-COLL-COORD.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: COLL-REQ                  **COLREQ.DRA**
C
C
C      ABSTRACT:
C
C           BEG-FUNC COLL-REQ;
C                COMMENT: ANALYST DISPLAYS AND FILLS IN COLLECTION
C                         REQUEST FORMS;
C                FORM G001.1;
C                THINK(5,10,);
C                ENTER(G001#3 = '$' ,G001#6 = '$',G001#43 = C001#40);
C                THINK(120,180,);
C                COMMENT: AFTER COMPLETING THE REQUEST FORM, THE ANALYST
C                         STORES IT IN THE DATABASE;
C                ADDON G001.1;
C                COMMENT: THE ANALYST CAN NO LONGER CONTINUE ON THE CURRENT
C                         ASSESSMENT WITH THE INFO. AT HAND;
C           END-FUNC COLL-REQ.
C      INPUT:
C                     C1A040 - target country code for request
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: MISSION-UPDATE          **MSNUPD.DRA**
C
C
C      ABSTRACT:
C              BEG-FUNC MISSION-UPDATE;
C                 RETRIEVE(H001.1,H001#2 EQ '$');
C                 THINK(5,15,);
C                 MODIFY(H001#31 = 'WATCH FOR SAM');
C              END-FUNC MISSION-UPDATE.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: CHATTER-MAC     ***CHTMAC.FTN***
C
C
C      ABSTRACT:
C              BEG-FUNC CHATTER-MAC;
C                 INSERT(,,BLOCK = 'ANY MISSIONS IN THIS AREA?');
C                 THINK(5,10, );
C                 CHATTER MAC-ROUTE-THREAT-ASSESS;
C                 COMMENT: THE ANALYST WAITS FOR MAC TO RESPOND WITH MISSION-IDS
C                          FOR ALL OPERATIONS IN THE AREA;
C                 THINK(300,600, );
C                 SCAN;
C                 THINK(5,10, );
C                 DISPLAY;
C                 THINK(20,30, );
C              END-FUNC CHATTER-MAC.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: STORE-MAP               **STOMAP.FTN**
C
C
C       ABSTRACT:
C
C           BEG-FUNC STORE-MAP;
C               COMMENT:  STORE THE FULLY CONSTRUCTED MAP FOR
C                       LATER USE IN BRIEFINGS;
C               STORE;
C           END-FUNC STORE-MAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: EXAMINE-INDICATORS      **EXMIND.ORA**
C
C       IDENT:          G80-007.XX
C
C       TASK NAME:      ANY SCRIPT OR FUNCTION, LINK W/DHIL.OLB
C
C       AUTHOR:         , Mc2
C
C       ABSTRACT:
C             BEG-FUNC EXAMINE-INDICATORS;
C                COMMENT: THE ANALYST REQUESTS ALL MILITARY INDICATORS
C                         FOR THE COUNTRY OBSERVED;
C                QUERY(F001.1,F001#1 EQ C001#40 AND F001#21 EQ 'MIL');
C                THINK(10,15,);
C                PAGE;
C                THINK(10,15,);
C             END-FUNC EXAMINE-INDICATORS.
C
C       INPUT:
C                       C1A040 - country code
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ACTIVATE-INDICATORS     **ACTIND.ORA**
C
C
C       ABSTRACT:
C
C           BEG-FUNC ACTIVATE-INDICATORS;
C               COMMENT: UPDATE THE INDICATOR STATUS;
C               MODIFY(F001#24 = 'A');
C               COMMENT: THE ANALYST MAKES HARDCOPY FOR LATER REFERENCES;
C               HARDCOPY;
C               THINK(5,10,);
C           END-FUNC ACTIVATE-INDICATORS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C
C       PROCEDURE NAME:  PRESENT BRIEFING     PREBRF.ORA
C
C
C       ABSTRACT:
C               BEG-FUNC PRESENT-BRIEFING
C                 COMMENT: ANALYST DISPLAYS THE FULLY CONSTRUCTED MAP;
C                 RECALL;
C                 THINK(300,600,A);
C                 COMMENT: AS AN AIDE IN BRIEFING THE ASSESSMENT, THE
C                          ANALYST WILL IDENTIFY THE MISSION IN DANGER
C                          WITH LIGHTPEN/CURSUR AND DISPLAY INFO ABOUT
C                          THE ROUTE;
C                 AMP;
C                 THINK(300,600,A);
C               END-FUNC PRESENT-BRIEFING.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PREPARE-MSG              **PRENSG.ORA**
C
C
C       ABSTRACT:
C               BEG-FUNC PREPARE-MSG;
C                 COMMENT: ANALYST RETRIEVES MESSAGE FORMAT AND ENTERS
C                          APPROPRIATE DATA;
C                 FORM A001.1;
C                 THINK(5,10, );
C                 ENTER(A001#1 = '$',A001#71 = 'I',A001#190 = 'SAM WARNING',
C                       A001#200 = 'MESSAGE TEXT');
C                 THINK(30,60, );
C                 COMMENT: THE WARNING MESSAGE WILL BE ADDED TO THE DATABASE
C                          MESSAGE FILE,THE ANALYST WILL MAKE A PERSONAL COPY
C                          THEN ROUTE THE MESSAGE TO THE SUPERVISORS ALERT QUEUE;
C                 ADDON A001.1;
C                 HARDCOPY;
C                 THINK(5,10, );
C                 ROUTE(LOGIN = 'DIV-CHIEF',MSG-ID = A001#1,PRI = 'I',
C                       SUBJ = 'SAM WARNING');
C               END-FUNC PREPARE-MSG.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT-FLT-ROUTES          **PLTRTE.DRA**
C
C
C       ABSTRACT:
C               BEG-FUNC PLOT-FLT-ROUTES;
C                  COMMENT: THE ANALYST WILL REQUEST ALL MISSION INFORMATION
C                           TO OVERLAY ON PREVIOUSLY CONSTRUCTED MAP,
C                           BASED ON THE INFO. MAC SENT;
C                  SOMETIMES(.30,A,LOOP=3);
C                       BEG-SOME;
C                       RETRIEVE(H001.1,H001#2 EQ '#');
C                       THINK(120,300,A);
C                       COMMENT: PLOT THE DEPARTURE POINT AND TAG EACH ROUTE
C                                WITH ITS MISSION-ID;
C                       PLOT(,CLASS = 'MSN',);
C                       THINK(5,10,);
C                       INSERT(,,BLOCK = H001#2);
C                       COMMENT: THE ANALYST REQUESTS AND PLOTS ALL
C                                SEGMENT INFORMATION;
C                       SEARCH(I002.1,I002#70 EQ H001#2);
C                          BEG-SEARCH;
C                          THINK(5,10,);
C                          REVIEW;
C                          THINK(10,30,);
C                          SEG(,,TOP-LAT = I002#7,TOP-LON = I002#7A)
C                          END-SEARCH;
C                       END-SOME;
C       END-FUNC PLOT-FLT-ROUTES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: THINK
C
C
C       ABSTRACT:
C
C          THINK simulates an analyst mulling over a decision.  Because of
C       this, the result is divided by the alert status.  Obviously, the
C       analyst will think less time in a crisis or war situation.
C
C
C       INPUT:          MIN     - INTEGER;  Minimum peacetime wait
C                       MAX     - INTEGER;  Maximum wait at any time
C                       ALERT   - INTEGER;  Alert status (1,2,3)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ALERT
C
C
C       ABSTRACT:
C
C          ALERT will simulate an analyst displaying numbers and types
C       of messages on the alert queue.  The actual messages are not relevant.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SCAN
C
C
C       ABSTRACT:
C
C          SCAN simulates an analyst scanning through the list of alert
C       messages.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: STORE
C
C
C       ABSTRACT:
C
C            THE DHIL 'STORE' COMMAND IS USED TO SAVE THE CURRENT SCREEN
C       DISPLAY (MAP/RECORD) IN A TEMPORARY FILE FOR FAST RECALL AT A LATER
C       TIME.  ONLY THREE 'RECORDS' CAN BE STORED AT ANY TIME, AND THEY ARE
C       ACCESSED ON A LAST-IN FIRST-OUT BASIS VIA THE 'RECALL' COMMAND.
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: HARDCOPY                        $$HRDCPY.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'HARDCOPY' COMMAND IS USED TO FORMAT AND TRANSFER
C       ALL OF THE CURRENT DISPLAY TO A HARDCOPY DEVICE.
C
C       THE RTE COMMAND FORMAT IS:
C            HARDCOPY;
C       THE COMMAND WILL BE ISSUED AS A WAIT TIME FOR THE BENCHMARK
C       EXERCISES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: FORM                              $$FORM.FTN$$
C
C
C       ABSTRACT:
C
C            The FORM command is used to display the template
C       format for the specified userview.
C
C       The RTE command format is:
C            FORM <userview>;
C       This command is used to display a form which will then
C       have information ENTERed into it by the analyst (a sequence
C       of steps used to add a record to the database).  For testing
C       purposes this command will be executed as a WAIT time.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE - ROUTE A MESSAGE TO ANOTHER ANALYST
C
C
C       ABSTRACT:
C
C          ROUTE executes a wait time that simulates an analyst typing in
C       another analysts ID and sending him a message.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

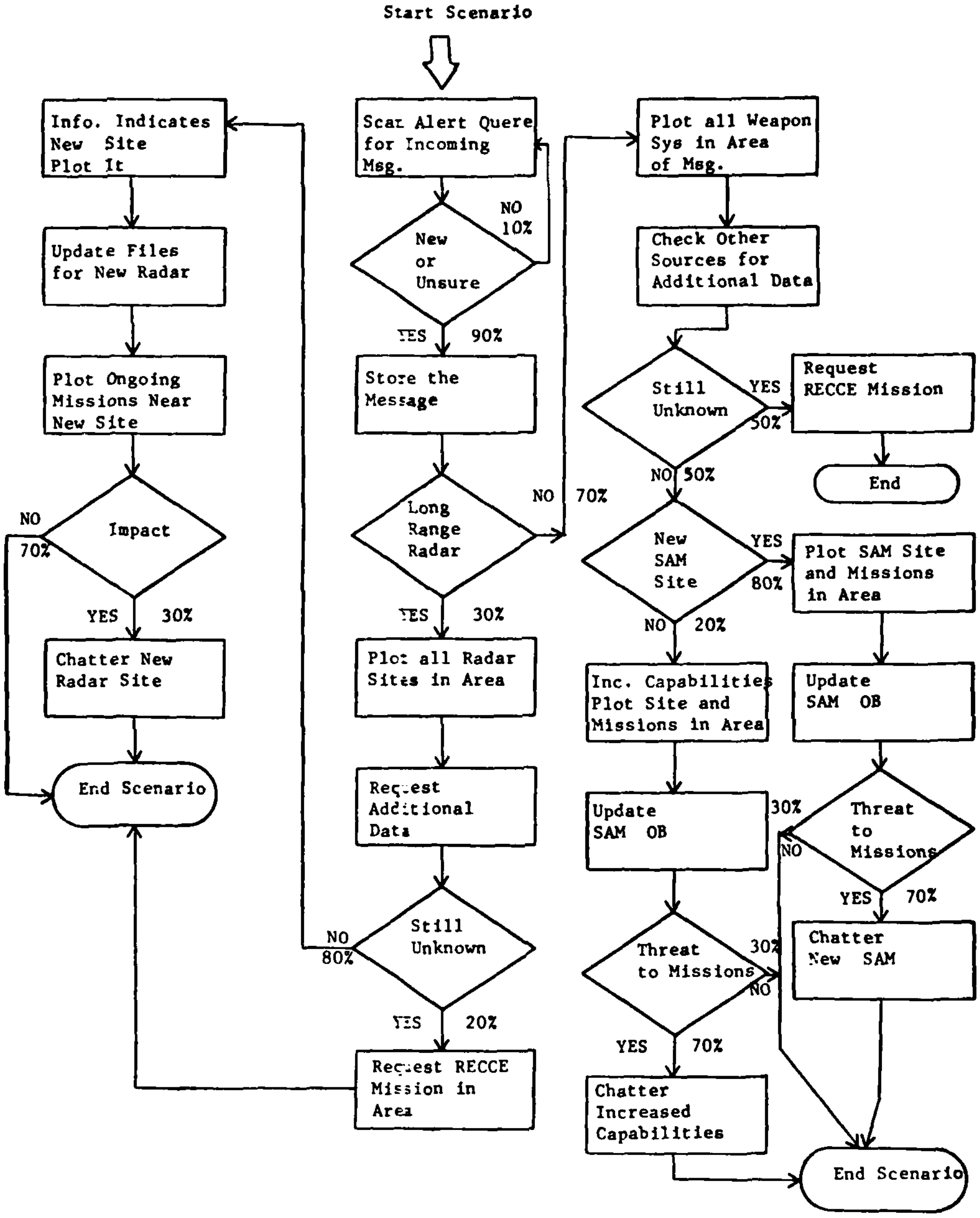


Figure C-5 Military - Systems - Analysis

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: MIL-SYS-ANAL                  **MILSYS.ORA**
C
C
C       ABSTRACT:
C
C            BEG-SCEN MIL-SYS-ANAL;
C            COMMENT:  ANALYST IS SCANNING ALERT QUEUE FOR INCOMING
C                      MESSAGES;
C            PERFORM REVIEW-INPUTS;
C            COMMENT:  AN INVESTIGATION IS WARRANTED IF THE MESSAGE
C                      CONTAINS NEW OR UNUSUAL SYSTEMS DATA;
C            SOMETIMES(.90,,);
C                 BEG-SOME;
C                 PERFORM STORE-MSG;
C                 COMMENT:  THE MESSAGE MAY PERTAIN TO SEARCH (LONG-RANGE)
C                           RADAR ONLY;
C                 SOMETIMES(.30,,);
C                      BEG-SOME;
C                      COMMENT:  THF ANALYST IDENTIFIES AND PLOTS ALL
C                                SYSTEMS IN THE AREA IDENTIFIED IN THE
C                                MESSAGE;
C                      PERFORM PLOT-RADAR-SITES;
C                      COMMENT:  AFTER STUDYING THE MAP, THE ANALYST
C                                DETERMINES THAT ADDITIONAL DATA IS
C                                REQUIRED TO EXPLAIN THE MESSAGE;
C                      THINK(10,15,);
C                      PERFORM CHECK-OTHER-SOURCES;
C                      COMMENT:  WHEN OTHER SOURCES CANNOT EXPLAIN THE
C                                MISSING INFORMATION, A FORMAL COLLECTION
C                                REQUEST BECOMES NECESSARY;
C                      SOMETIMES(.20,,);
C                           BEG-SOME;
C                           PERFORM SYS-COLL-REQ;
C                           PERFORM DONE-PROCESSING;
C                           END-SOME;
C                      COMMENT:  WHEN INFO. FROM OTHER SOURCES VERIFIES
C                                THE MESSAGE, THE ANALYST PROCEEDS BY
C                                ASSESSING THE IMPACT ON CURRENT OPNS;
C                      COMMENT:  INFO. INDICATES A NEW SITE;
C                      PERFORM PLOT-NEW-SITE;
C                      PERFORM OVERLAY-MSNS;
C                      PERFORM UPDATE-RADAR-INFO;
C                      SOMETIMES(.30,,);
C                           BEG-SOME;
C                           PERFORM CHATTER-NEW-SITE;
C                           END-SOME;
C                      PERFORM DONE-PROCESSING;
C                      END-SOME;
C                 COMMENT:  IF WE GOT HERE...
C                 COMMENT:  THE MESSAGE PERTAINS TO A PARTICULAR
C                           WEAPON SYSTEM;
C                 COMMENT:  THE ANALYST IDENTIFIES AND PLOTS ALL
C                           SYSTEMS IN THE AREA IDENTIFIED IN THE MSG;
C                 PERFORM PLOT-SAM-SITES;
C                 COMMENT:  AFTER STUDYING THE MAP, THE ANALYST DETERMINES
C                           THAT ADDITIONAL DATA IS REQUIRED TO EXPLAIN THE
C                           MESSAGE;
```

```
C               THINK(10,15,);
C               PERFORM CHECK-OTHER-SOURCES;
C               COMMENT:  WHEN OTHER SOURCES CANNOT EXPLAIN THE MISSING
C                         INFORMATION, A FORMAL COLLECTION REQUEST
C                         BECOMES NECESSARY;
C               SOMETIMES(.50,,);
C                    BEG-SOME;
C                    PERFORM SYS-COLL-REQ;
C                    PERFORM DONE-PROCESSING;
C                    END-SOME;
C               COMMENT:  WHEN INFO. FROM OTHER SOURCES VERIFIES THE
C                         MESSAGE, THE ANALYST PROCEEDS BY ASSESSING
C                         THE IMPACT ON CURRENT OPNS;
C               SOMETIMES(.80,,);
C                    BEG-SOME;
C                    COMMENT:  INFO. INDICATES A NEW SITE;
C                    PERFORM PLOT-NEW-SAM;
C                    PERFORM OVERLAY-MSNS;
C                    PERFORM UPDATE-SAM-INFO;
C                    SOMETIMES(.30,,);
C                         BEG-SOME;
C                         PERFORM CHATTER-NEW-SAM;
C                         END-SOME;
C                    PERFORM DONE-PROCESSING;
C                    END-SOME;
C               COMMENT:  IF WE GOT HERE...
C               COMMENT:  INFO. INDICATES INCREASED CAPABILITIES;
C               PERFORM PLOT-INC-CAP;
C               PERFORM OVERLAY-MSNS;
C               PERFORM UPDATE-SYS-DATA;
C               SOMETIMES(.30,,);
C                    BEG-SOME;
C                    PERFORM CHATTER-INC-CAP;
C                    END-SOME;
C               END-SOME;
C          END-SCEN MIL-SYS-ANAL.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: REVIEW-INPUTS           $$REVINP.DRA$$
C
C
C      ABSTRACT:
C
C          BEG-FUNC REVIEW-INPUTS;
C               COMMENT: DISPLAY NUMBER AND KIND OF EACH ENTRY IN QUEUE;
C               ALERT;
C               THINK(10,30,A);
C               COMMENT: DISPLAY SHORT TITLE - ALL QUEUED ALERTS;
C               SCAN;
C               THINK(10,30,A);
C               COMMENT: SELECTIVELY DISPLAY AND READ A MESSAGE FROM
C                        THOSE ROUTED THROUGH THE ALERT QUEUE;
C               RETRIEVE (A001.1, A001#1 EQ '$');
C               THINK(30,90,A);
C               PAGE;
C               THINK(30,90,A);
C          END-FUNC  REVIEW-INPUTS.
C
C      CALLED BY:     WATCH-FUNCTION
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: STORE-MSG                $$STOMSG.FTN$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC STORE-MSG;
C                 COMMENT:  STORE THE INPUT MESSAGE IN A TEMPORARY
C                           PERSONAL FILE FOR LATER USE;
C                 STORE;
C            END-FUNC STORE-MSG.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT-SAM-SITES           $$PLOTSS.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC PLOT-SAM-SITES;
C                 COMMENT:  THE ANALYST BUILDS A MAP OF THE AREA
C                           REFERENCED IN THE MESSAGE;
C                 PERFORM BUILD-AREA-MAP;
C                 COMMENT:  THE ANALYST SEARCHES FOR AND PLOTS ALL
C                           SAM SITES (OF THE IDENTIFIED TYPE)
C                           KNOWN TO BE IN THE AREA;
C                 CSEARCH(E002.1,E002#140 EQ 'SAM' AND E002#142 EQ '$',
C                         LAT = '$', LON = '$', RANGE = 500KM);
C                      BEG-SEARCH;
C                      THINK(2,5,);
C                      COMMENT:  ANALYST QUICKLY SCANS EACH RECORD
C                                BEFORE PLOTTING LOCATION ONTO MAP;
C                      REVIEW;
C                      THINK(10,15,);
C                      PLOT(,CLASS = 'SAM',);
C                      THINK(2,5,);
C                      COMMENT:  ANALYST INDICATES SAM RANGE ON MAP;
C                      CIRCLE(,RANGE = '$');
C                      END-SEARCH;
C            END-FUNC PLOT-SAM-SITES.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CHECK-OTHER-SOURCES       **CHKOS.FTN**
C
C
C       ABSTRACT:
C            BEG-FUNC CHECK-OTHER-SOURCES;
C                 COMMENT:  THE ANALYST WILL RECALL THE MESSAGE
C                           BEING INVESTIGATED, AND SEND A 'COPY'
C                           OF THE MESSAGE TO THE WATCH OFFICER
C                           AND COLLECTION COORDINATOR -- REQUESTING
C                           ANY INFORMATION THEY MIGHT HAVE;
C                 RECALL;
C                 ENTER(A001#190 = 'ANYONE KNOW ANYTHING ABOUT THIS?');
C                 CHATTER WATCH-OFFICER, COLLECTION-COORDINATOR;
C                 THINK(300,600,);
C                 COMMENT:  THE ANALYST WAITS FOR SOME REPLY;
C                 SCAN;
C                 THINK(5,15,);
C                 COMMENT:  ANALYST HAS RECEIVED A CHATTER OF THE MESSAGE
C                           WITH A RESPONSE;
C                 DISPLAY;
C                 THINK(10,30,);
C                 STORE;
C            END-FUNC CHECK-OTHER-SOURCES.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SYS-COLL-REQ            **SYSCOL.ORA**
C
C
C       ABSTRACT:
C
C            BEG-FUNC SYS-COLL-REQ;
C                 COMMENT:  THE ANALYST DISPLAYS AND FILLS IN COLLECTION
C                           REQUEST FORM;
C                 FORM G001.1;
C                 THINK(5,10,);
C                 ENTER(G001#3 = '$', G001#6 = '$', G001#43 = '$');
C                 THINK(120,180,);
C                 COMMENT:  AFTER COMPLETING THE REQUEST FORM, THE
C                           ANALYST STORES IT IN THE DATABASE;
C                 ADDON G001.1;
C                 COMMENT:  CURRENT OPNS SHOULD BE NOTIFIED THAT THE
C                           ASSESSMENT COULD NOT BE COMPLETED --
C                           SEND A COPY OF THE MESSAGE;
C                 RECALL;
C                 ENTER(A001#71 = 'NOTIFY OPNS - CAN'T VERIFY WHAT'S
C                      OUT THERE');
C                 THINK(5,10,);
C                 CHATTER DIV-CHIEF;
C                 COMMENT:  ANALYST CAN NO LONGER PROCEED WITH CURRENT
C                           ASSESSMENT;
C            END-FUNC SYS-COLL-REQ.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: PLOT-NEW-SITE          $$PLTNSI.FTN$$
C
C
C      ABSTRACT:
C
C          BEG-FUNC PLOT-NEW-SITE;
C              COMMENT:  ANOTHER SOURCE HAS IDENTIFIED THE INSTALLATION
C                        AND COORDINATES OF THE NEW RADAR SITE.  THE
C                        ANALYST PLOTS THIS INFORMATION AND CONTINUES
C                        ASSESSMENT;
C              INSERT(CLASS = 'LRR',,BLOCK = 'NEW SITE');
C              THINK(5,15,);
C              CIRCLE(,RANGE = 350KM);
C          END-FUNC PLOT-NEW-SITE.
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: OVERLAY-MSNS           $$OVMSNS.ORA$$
C
C
C      ABSTRACT:
C
C          BEG-FUNC OVERLAY-MSNS;
C              PERFORM CHATTER-MAC;
C              COMMENT:  MAC WILL EVALUATE THE AREA AND RETURN THE
C                        MISSION-IDS FOR ALL SCHEDULED MISSIONS IN
C                        THE AREA;
C              COMMENT:  THE ANALYST WILL REQUEST AND PLOT ALL
C                        INFORMATION ON THESE MISSIONS;
C              PERFORM PLOT-FLT-ROUTES;
C          END-FUNC OVERLAY-MSNS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: UPDATE-RADAR-INFO      $$UPDRI.ORA$$
C
C
C      ABSTRACT:
C
C          BEG-FUNC UPDATE-RADAR-INFO;
C              COMMENT:  THE ANALYST MUST UPDATE THE DATABASE
C                        TO REFLECT THE NEW RADAR SITE;
C              RETRIEVE(E002.3,E002$1 EQ '$');
C              THINK(10,20,);
C              MODIFY(E002$143 = '$');
C          END-FUNC UPDATE-RADAR-INFO.
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CHATTER-NEW-SITE          $$CHTNSI.FTN$$
C
C       ABSTRACT:
C
C            BEG-FUNC CHATTER-NEW-SITE;
C                 INSERT(,,BLOCK = 'NOTIFY OPNS -- NEW RADAR SITE HAS
C                           BEEN VERIFIED');
C                 THINK(5,10,);
C                 CHATTER DIV-CHIEF;
C                 BREAK;
C            END-FUNC CHATTER-NEW-SITE.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT-SAM-SITES            $$PLOTSS.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC PLOT-SAM-SITES;
C                 COMMENT:  THE ANALYST BUILDS A MAP OF THE AREA
C                           REFERENCED IN THE MESSAGE;
C                 PERFORM BUILD-AREA-MAP;
C                 COMMENT:  THE ANALYST SEARCHES FOR AND PLOTS ALL
C                           SAM SITES (OF THE IDENTIFIED TYPE)
C                           KNOWN TO BE IN THE AREA;
C                 CSEARCH(E002.1,E002#140 EQ 'SAM' AND E002#142 EQ '$',
C                         LAT = '$', LON = '$', RANGE = 500KM);
C                      BEG-SEARCH;
C                      THINK(2,5,);
C                      COMMENT:  ANALYST QUICKLY SCANS EACH RECORD
C                                BEFORE PLOTTING LOCATION ONTO MAP;
C                      REVIEW;
C                      THINK(10,15,);
C                      PLOT(,CLASS = 'SAM',);
C                      THINK(2,5,);
C                      COMMENT:  ANALYST INDICATES SAM RANGE ON MAP;
C                      CIRCLE(,RANGE = '$');
C                      END-SEARCH;
C            END-FUNC PLOT-SAM-SITES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT-NEW-SAM              $$PLTNSA.FTN$$
C       ABSTRACT:
C
C            BEG-FUNC PLOT-NEW-SAM;
C                 COMMENT:  ANOTHER SOURCE HAS IDENTIFIED THE COORDINATES
C                           OF A NEW SAM SITE.  THE ANALYST PLOTS THIS
C                           INFORMATION AND CONTINUES ASSESSMENT;
C                 INSERT(CLASS = 'SAM',,BLOCK = 'NEW SAM');
C                 THINK(5,15,);
C                 CIRCLE(,RANGE = '$');
C            END-FUNC PLOT-NEW-SAM.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: OVERLAY-MSNS              $$OVMSNS.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC OVERLAY-MSNS;
C                 PERFORM CHATTER-MAC;
C                 COMMENT:  MAC WILL EVALUATE THE AREA AND RETURN THE
C                           MISSION-IDS FOR ALL SCHEDULED MISSIONS IN
C                           THE AREA;
C                 COMMENT:  THE ANALYST WILL REQUEST AND PLOT ALL
C                           INFORMATION ON THESE MISSIONS;
C                 PERFORM PLOT-FLT-ROUTES;
C            END-FUNC OVERLAY-MSNS.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: UPDATE-SAM-INFO           $$UPDSAM.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC UPDATE-SAM-INFO;
C                 COMMENT:  THE ANALYST MUST UPDATE THE DATABASE
C                           TO REFLECT THE NEW SAM SITE;
C                 RETRIEVE(E002.1,E002#1 EQ '$');
C                 THINK(10,20,);
C                 MODIFY(E002#143 = '$');
C            END-FUNC UPDATE-SAM-INFO.
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: CHATTER-NEW-SAM          $$CHTNSA.FTN$$
C
C
C      ABSTRACT:
C
C           BEG-FUNC CHATTER-NEW-SAM;
C                INSERT(,,BLOCK = 'WARN OPNS - NEW SAM SITE HAS
C                         BEEN VERIFIED');
C                THINK(5,10,);
C                CHATTER DIV-CHIEF;
C                BREAK;
C           END-FUNC CHATTER-NEW-SAM.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: CHATTER-INC-CAP          $$CHATIC.FTN$$
C
C
C      ABSTRACT:
C
C           BEG-FUNC CHATTER-INC-CAP;
C                INSERT(,,BLOCK = 'NOTIFY OPNS -- INCREASED CAPABILITY
C                         FOR THIS SYSTEM HAS BEEN VERIFIED');
C                THINK(5,10,);
C                CHATTER DIV-CHIEF;
C           END-FUNC CHATTER-INC-CAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: UPDATE-SYS-DATA          $$UPSYSD.ORA$$
C
C
C      ABSTRACT:
C
C           BEG-FUNC UPDATE-SYS-DATA;
C                COMMENT:  ANALYST RETRIEVES THE RECORD FOR THIS SYSTEM;
C                RETRIEVE(D001.1,D001#1 EQ 'SAM' AND D001#2 EQ '$');
C                THINK(5,10,);
C                COMMENT:  ANALYST CORRECTS DATA BASED ON CONTENT OF
C                          MESSAGE AND EVALUATION;
C                MODIFY(D001#40 = '$');
C           END-FUNC UPDATE-SYS-DATA.
C
C      CALLED BY:      MIL-SYS-ANAL
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCLCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: THINK
C
C
C       ABSTRACT:
C
C          THINK simulates an analyst mulling over a decision.  Because of
C       this, the result is divided by the alert status.  Obviously, the
C       analyst will think less time in a crisis or war situation.
C
C
C       INPUT:          MIN     - INTEGER;  Minimum peacetime wait
C                       MAX     - INTEGER;  Maximum wait at any time
C                       ALERT   - INTEGER;  Alert status (1,2,3)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ALERT
C
C
C       ABSTRACT:
C
C          ALERT will simulate an analyst displaying numbers and types
C       of messages on the alert queue.  The actual messages are not relevant.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SCAN
C
C
C       ABSTRACT:
C
C          SCAN simulates an analyst scanning through the list of alert
C       messages.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: STORE
C
C
C       ABSTRACT:
C
C            THE DHIL 'STORE' COMMAND IS USED TO SAVE THE CURRENT SCREEN
C       DISPLAY (MAP/RECORD) IN A TEMPORARY FILE FOR FAST RECALL AT A LATER
C       TIME.  ONLY THREE 'RECORDS' CAN BE STORED AT ANY TIME, AND THEY ARE
C       ACCESSED ON A LAST-IN FIRST-OUT BASIS VIA THE 'RECALL' COMMAND.
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: BUILD-AREA-MAP  ***BLDMAP.FTN***
C
C       IDENT:          GBO-007.XX
C
C       TASK NAME:      ANY SCRIPT OR FUNCTION, LINK W/DHIL.OLB
C
C       AUTHOR:         , Mc2
C
C       ABSTRACT:
C               BEG-FUNC BUILD-AREA-MAP;
C                  COMMENT: ANALYST BUILDS A MAP OF THE AREA REFERRED TO IN
C                           THE MESSAGE;
C                  MAP(LAT = '$' ,LON = '$', );
C                  THINK(2,5, );
C               END-FUNC BUILD-AREA-MAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CIRCLE                  **CIRCLE.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'CIRCLE' COMMAND IS USED TO DRAW A CIRCLE WITH THE
C       SPECIFIED RANGE AROUND AN ITEM THAT HAS BEEN IDENTIFIED ON THE
C       CURRENT MAP DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       CIRCLE([<fieldname><relop><value>|BLOCK<relop><value>],
C             RANGE = <range>);
C
C       PARAMETERS ARE:
C       <fieldname><relop><value> - OPTIONAL PARAMETER WHICH SPECIFIES THE
C             ITEM ON DISPLAY WHICH SHOULD BE CIRCLED.
C       BLOCK<relop><value> - OPTIONAL PARAMETER WHICH IDENTIFIES THE ITEM
C             WHICH SHOULD BE CIRCLED THROUGH IT'S DATA BLOCK VALUE.
C       RANGE = <range> - REQUIRED PARAMETER WHICH SPECIFIES THE RADIUS OF
C             THE CIRCLE TO BE DRAWN. (VALID UNITS ARE NM AND KM, DEFAULT
C             TO KM IF NOT SPECIFIED)
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT                         **PLOT.FTN**
C
C
C       ABSTRACT:
C
C       THE DHIL 'PLOT' COMMAND IS USED TO DISPLAY DBMS ITEMS ONTO A
C       PREVIOUSLY GENERATED MAP BACKGROUND.
C
C       THE RTE COMMAND FORMAT IS:
C       PLOT([<userview>,<fieldname> <relop> <fieldvalue> [<logop>
C             <fieldname> <relop> <fieldvalue>]],CLASS = <symbol>,
C             [BLOCK|NOBLOCK]);
C
C       PARAMETERS ARE:
C       <userview>,<fieldname>...<fieldvalue> - OPTIONAL SELECTION LIST
C               WHICH WHEN SPECIFIED CAUSES A DATABASE ACCESS FOR ALL RECORDS
C               THAT MEET THE CRITERIA TO BE PLOTTED ONTO THE MAP.
C               IF NOT SPECIFIED, PLOT THE INF   ASSOCIATED WITH THE RECORD
C               CURRENTLY BEING DISPLAYED.
C       CLASS - REQUIRED PARAMETER WHICH INDICATES THE CLASSIFICATION OF
C               SYMBOLIC DATA TO BE PRESENTED ON THE MAP
C       BLOCK|NOBLOCK - OPTIONAL PARAMETER WHICH SPECIFIES WHETHER A 'DATA
C               BLOCK/TAG' IS TO BE APPENDED TO EACH SYMBOL GENERATED FOR
C               DISPLAY.  BLOCK IS THE DEFAULT.
C
C       NOTE: THIS IS THE ONLY DHIL COMMAND THAT WILL EITHER BE EXECUTED AS
C             A 'WAIT' TIME (WHEN ISSUED WITH NO SELECTION LIST), OR MAKE A
C             DATABASE ACCESS (WHEN ISSUED WITH A SELECTION LIST).
C
C       *********************************************************************
C       *********************************************************************
C       ****THIS 'PLOT' FUNCTION SHOULD ONLY BE CALLED BY FUNCTIONS WHICH****
C       ****ARE ISSUING A DHIL PLOT COMMAND WITHOUT A SELECTION LIST*********
C       *********************************************************************
C       *********************************************************************
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: RECALL                       **RECALL.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'RECALL' COMMAND IS USED TO ACCESS A MAP/RECORD
C       WHICH HAS BEEN STOREd IN A TEMPORARY FILE.
C
C       THE RTE WILL EXECUTE THIS COMMAND AS A 'WAIT' TIME
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ENTER                    **ENTER.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'ENTER' COMMAND IS USED TO SIMULATE THE ANALYST
C       ENTERING DATA INTO A PREVIOUSLY DISPLAYED FORM.  ONLY THE CONTENTS
C       OF THE DISPLAY ARE MODIFIED -- THIS COMMAND HAS NO EFFECT ON THE
C       DATABASE.  SUBSEQUENT PRIMITIVE DIRECTIVES CONTROL THE DISPOSITION
C       OF THE DATA.
C
C       THE RTE COMMAND FORMAT IS:
C       ENTER(<fieldname> = <new value>[,<fieldname> = <new value>]);
C
C       PARAMETERS ARE:
C            <fieldname> - THE NAME OF THE FIELD RECEIVING DATA
C            <new value> - VALUE TO BE ASSIGNED TO RECEIVING FIELD
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CHATTER                  **CHTTER.FTN**
C
C       ABSTRACT:
C
C            THE DHIL 'CHATTER' COMMAND IS USED TO SEND A COPY OF THE
C       CURRENT TERMINAL DISPLAY TO ANOTHER ANALYST THROUGH THE ALERT
C       QUEUE.
C
C       THE RTE COMMAND FORMAT IS:
C       CHATTER <receiving LOGIN identifier> [,<receiving LOGIN identifier>];
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: DISPLAY                  **DISPLY.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'DISPLAY' COMMAND IS USED TO DISPLAY AN ALERT
C       QUEUE ENTRY TO THE TERMINAL.   THE RTE FORMAT IS:
C
C       DISPLAY [<alert number>][INTO FORM <userview>];
C
C       BOTH ARE OPTIONAL PARAMETERS, WHERE;
C       <alert number> - THE NUMBER OF THE ALERT QUEUE ENTRY TO BE
C                        DISPLAYED.  IF NOT SPECIFIED, DEFAULT TO
C                        HIGHEST PRIORITY ALERT QUEUE ENTRY.
C       INTO FORM <userview> - PARAMETER USED BY THE UPDATE SCENARIO
C                        TO BISPLAY AN ALERT QUEUE ENTRY INTO THE
C                        SPECIFIED <userview> FORMAT.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: FORM                      **FORM.FTN**
C
C
C       ABSTRACT:
C
C            The FORM command is used to display the template
C       format for the specified userview.
C
C       The RTE command format is:
C            FORM <userview>;
C       This command is used to display a form which will then
C       have information ENTERed into it by the analyst (a sequence
C       of steps used to add a record to the database).  For testing
C       purposes this command will be executed as a WAIT time.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CHATTER-MAC     ***CHTMAC.FTN***
C
C       ABSTRACT:
C               BEG-FUNC CHATTER-MAC;
C                  INSERT(,,BLOCK = 'ANY MISSIONS IN THIS AREA?");
C                  THINK(5,10,);
C                  CHATTER MAC-ROUTE-THREAT-ASSESS;
C                  COMMENT: THE ANALYST WAITS FOR MAC TO RESPOND WITH
C                           MISSION-IDs FOR ALL OPERATIONS IN THE AREA;
C                  THINK(300,600,);
C                  SCAN;
C                  THINK(5,10,);
C                  DISPLAY;
C                  THINK(20,30,);
C               END-FUNC CHATTER-MAC.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: PLOT-FLT-ROUTES          ##PLTRTE.DRA##
C
C
C      ABSTRACT:
C              BEG-FUNC PLOT-FLT-ROUTES;
C                 COMMENT: THE ANALYST WILL REQUEST ALL MISSION INFORMATION
C                          TO OVERLAY ON PREVIOUSLY CONSTRUCTED MAP,
C                          BASED ON THE INFO. MAC SENT;
C                 SOMETIMES(.30,A,LOOP=3);
C                      BEG-SOME;
C                      RETRIEVE(H001.1,H001#2 EQ '$');
C                      THINK(120,300,A);
C                      COMMENT: PLOT THE DEPARTURE POINT AND TAG EACH ROUTE
C                               WITH ITS MISSION-ID;
C                      PLOT(,CLASS = 'MSN',);
C                      THINK(5,10,);
C                      INSERT(,,BLOCK = H001#2);
C                      COMMENT: THE ANALYST REQUESTS AND PLOTS ALL
C                               SEGMENT INFORMATION;
C                      SEARCH(I002.1,I002#70 EQ H001#2);
C                         BEG-SEARCH;
C                         THINK(5,10,);
C                         REVIEW;
C                         THINK(10,30,);
C                         SEG(,,TOP-LAT = I002#7,TOP-LON = I002#7A)
C                         END-SEARCH;
C                      END-SOME;
C      END-FUNC PLOT-FLT-ROUTES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: INSERT                   ##INSERT.FTN##
C
C
C      ABSTRACT:
C
C           THE DHIL 'INSERT' COMMAND IS USED TO ADD A NEW ITEM TO THE
C      SPECIFIED LOCATION ON THE CURRENT MAP DISPLAY WITH A DATA BLOCK
C      APPENDED TO IT, OR TO ASSOCIATE A DATA BLOCK WITH AN ITEM ALREADY
C      ON THE CURRENT MAP DISPLAY.
C
C      THE RTE COMMAND FORMAT IS:
C      INSERT([CLASS = <symbol>],[LOC = <value>],BLOCK = <value>);
C
C      PARAMETERS ARE:
C      CLASS = <symbol> - OPTIONAL PARAMETER WHICH SPECIFIES THE SYMBOLOGY
C                         CLASSIFICATION OF THE ITEM BEING ADDED TO THE
C                         DISPLAY.
C      LOC = <value>    - OPTIONAL PARAMETER WHICH SPECIFIES THE LOCATION
C                         VALUE OF THE ITEM BEING ADDED TO THE DISPLAY.
C                         IF NOT GIVEN, DEFAULT TO THE CURRENT LOCATION
C                         OF THE CURSOR/LIGHT PEN.
C      BLOCK = <value>  - REQUIRED PARAMETER WHICH SPECIFIES THE DATA BLOCK
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: BUILD-AREA-MAP  ***BLDMAP.FTN***
C
C
C       ABSTRACT:
C               BEG-FUNC BUILD-AREA-MAP;
C                  COMMENT: ANALYST BUILDS A MAP OF THE AREA REFERRED TO IN
C                           THE MESSAGE;
C                  MAP(LAT = '$' ,LON = '$', );
C                  THINK(2,5, );
C               END-FUNC BUILD-AREA-MAP.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

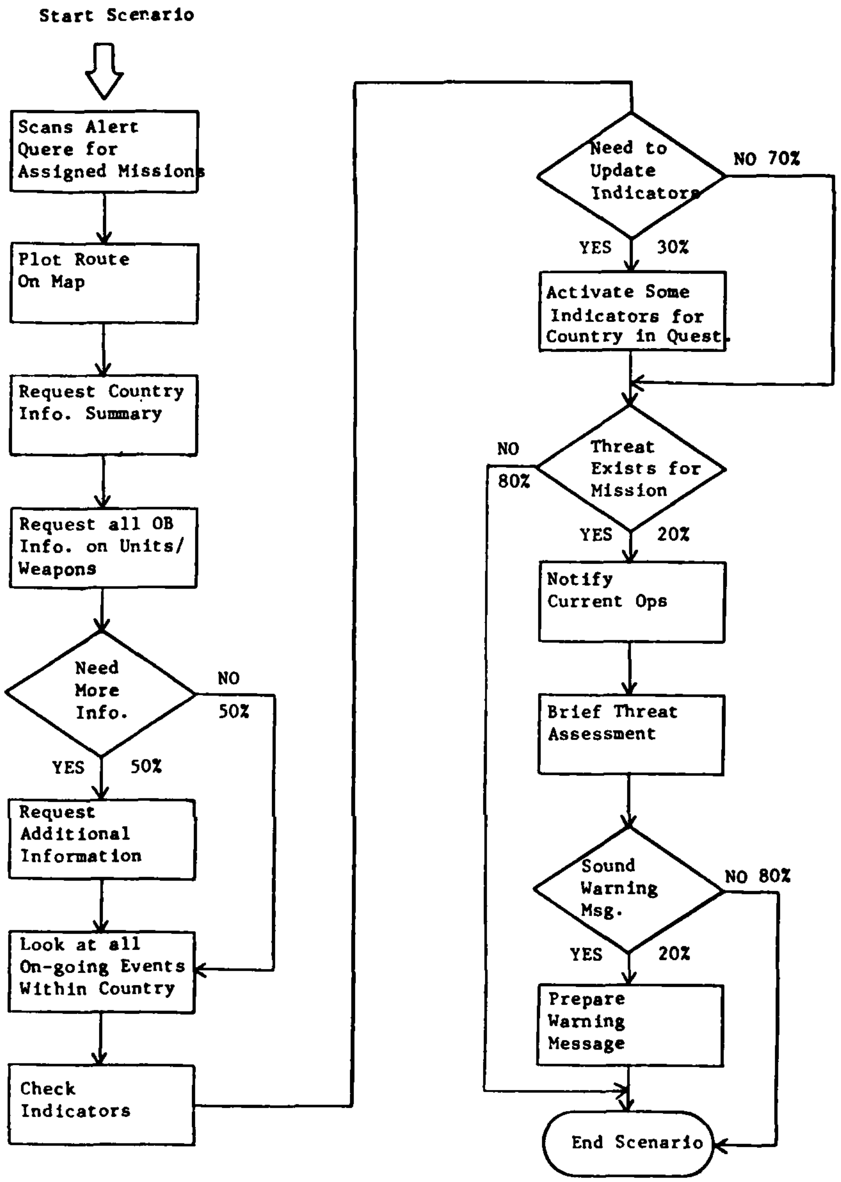

Start Scenario
Scans Alert Quere for Assigned Missions
Plot Route On Map
Request Country Info. Summary
Request all OB Info. on Units/ Weapons
Need More Info.
NO 50%
YES 50%
Request Additional Information
Look at all On-going Events Within Country
Check Indicators
Need to Update Indicators
NO 70%
YES 30%
Activate Some Indicators for Country in Quest.
Threat Exists for Mission
NO 80%
YES 20%
Notify Current Ops
Brief Threat Assessment
Sound Warning Msg.
NO 80%
YES 20%
Prepare Warning Message
End Scenario

Figure C-6 MAC Route Assessment

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: MAC-ROUTE-ASSESSMENT      $$MACRTE.ORA$$
C
C
C       AUTHOR:         , Mc2
C
C       ABSTRACT:
C               BEG-SCEN MAC-ROUTE-ASSESSMENT;
C                  COMMENT: ANALYST SCANS ALERT QUEUE FOR ASSIGNED MISSIONS;
C                  PERFORM SCAN-QUEUE;
C                  COMMENT: ANALYST RETRIEVES MISSION AND ROUTE DESCRIPTION
C                           INFORMATION AND PLOTS ON GRAPHICS TERMINAL;
C                  PERFORM PLOT-MSN;
C                  COMMENT: ANALYST REQUESTS COUNTRY SUMMARY INFO.;
C                  PERFORM COUNTRY-INFO;
C                  COMMENT: ANALYST REQUESTS ALL OB INFO TO LOCATE
C                           UNITS/WEAPON SYSTEMS FOR COUNTRY INVOLVED;
C                  PERFORM SEARCH-OB;
C                  COMMENT: ANALYST MAY NEED MORE INFO ABOUT A PARTICULAR
C                           WEAPON SYSTEM'S CAPABILITIES;
C                  SOMETIMES(.50, ,LOOP = 3);
C                       BEG-SOME;
C                       PERFORM FIND-SYS-CAP;
C                       END-SOME;
C                  COMMENT: ANALYST REQUESTS INFO. FOR ALL ON-GOING EVENTS
C                           WITHIN THE COUNTRY IN QUESTION;
C                  PERFORM SEARCH-EVENTS;
C                  COMMENT: ANALYST REQUESTS ALL INDICATORS FOR
C                           COUNTRY IN QUESTION;
C                  PERFORM QUERY-INDICATORS;
C                  COMMENT: ACTIVATE APPROPRIATE INDICATORS;
C                  SOMETIMES(.30,A, );
C                       BEG-SOME;
C                       PERFORM ACTIVATE-INDICATORS;
C                       END-SOME;
C                  COMMENT: BASED ON INFORMATION RETRIEVED, THE ANALYST
C                           DETERMINES IF A POSSIBLE THREAT EXISTS
C                           FOR THIS MISSION;
C                  SOMETIMES(.20,A, );
C                       BEG-SOME;
C                       COMMENT: SITUATION MAY REQUIRE NOTIFICATION
C                                TO CURRENT OPNS;
C                       PERFORM NOTIFY-OPNS;
C                       COMMENT: IF THREAT EXISTS, PRESENT BRIEFING;
C                       PERFORM BRIEF-ASSESSMENT;
C                       COMMENT: IF THREAT EXISTS, IT MAY BE NECESSARY TO ISSUE
C                                WARNING MESSAGE;
C                       SOMETIMES(.20, ,);
C                          BEG-SOME;
C                          PERFORM PREPARE-MSG;
C                          END-SOME;
C                       END-SOME;
C       END-SCEN MAC-ROUTE-ASSESSMENT.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C
C       PROCEDURE NAME: SCAN-QUEUE              ***SCHQUE.FTN***
C
C
C       ABSTRACT:
C               BEG-FUNC SCAN-QUEUE;
C                  COMMENT: WHEN AN OPERATION HAS BEEN SCHEDULED, MAC IS CALLE
C                           UPON TO EVALUATE THE DESIGNATED ROUTE;
C                  COMMEN": THE ANALYST IS SEARCHING THROUGH THE ALERT QUEUE
C                           TO FIND MISSIONS WHICH HAVE BEEN ASSIGNED FOR
C                           EVALUATION;
C                  ALERT;
C                  THINK(10,30,A);
C                  SCAN;
C                  THINK(10,30,A);
C                  DISPLAY;
C                  THINK(30,90,A);
C               END-FUNC SCAN-QUEUE.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT-MSN                **PLTMSN.ORA**
C
C
C       ABSTRACT:
C               BEG-FUNC PLOT-MSN;
C                  COMMENT: REQUEST AND READ THE MISSION DATA;
C                       RETRIEVE(H001.1,H001$2 EQ '$');
C                       THINK(10,30,A);
C                       COMMENT: CREATE A MAP BACKGROUND OF THE AREA;
C                       MAP(LOC = H001$7);
C                       THINK(2,5, );
C                       COMMENT: PLOT THE MISSION DEPARTURE POINT;
C                       PLOT(,CLASS = 'MSN',);
C                       INSERT( , ,BLOCK = H001$2);
C                       THINK(5,10, );
C                       COMMENT: THE ANALYST REQUESTS INFO. ON ALL ROUTE
C                                SEGMENTS, AND OVERLAYS ONTO MAP;
C                       SEARCH(I002.1,I002$70 EQ H001$2);
C                          BEG-SEARCH;
C                          THINK(2,5, );
C                          REVIEW;
C                          THINK(10,30, );
C                          SEG( , ,TO-LAT = I002$7,TO-LON = I002$7A)
C                          END-SEARCH;
C                       END-SOME;
C       END-FUNC PLOT-MSN.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: COUNTRY-INFO             $$CONINF.ORA$$
C
C
C       ABSTRACT:
C
C            BEG-FUNC COUNTRY-INFO;
C                 RETRIEVE(E001.1,E001#2 EQ H001#7);
C                 THINK(10,30,A);
C            END-FUNC COUNTRY-INFO.
C
C       CALLED BY:      MAC-ROUTE-ASSESSMENT
C
C       CALLS:          THINK, OSQL, OBIND, ODFINN, OEXEC, OFETCH
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SEARCH-OB                      $$SRCHOB.ORA$$
C
C
C       ABSTRACT:
C               BEG-FUNC SEARCH-OB;
C                  COMMENT: REQUEST ALL UNITS WITHIN 500KM OF ROUTE;
C                  CSEARCH(C001.1,LAT = '$',LON = '$',RANGE = 500KM);
C                       BEG-SEARCH;
C                       THINK(2,5, );
C                       REVIEW;
C                       THINK(30,90,A);
C                       COMMENT: IF LOCATION OF UNIT AND WEAPON SYSTEMS HAVE
C                                IMPACT ON MISSION ASSESSMENT -- PLOT;
C                       SOMETIMES(.50,A, );
C                          BEG-SOME;
C                          PLOT(, CLASS = '$');
C                          THINK(2,5, );
C                          INSERT(, ,BLOCK = C001#111B);
C                          THINK(2,5, );
C                          END-SOME;
C                       END-SEARCH;
C               END-FUNC SEARCH-OB.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: FIND-SYS-CAPABILITIES    $$FNDCAP.ORA$$
C
C
C      ABSTRACT:
C           BEG-FUNC FIND-SYS-CAP;
C                COMMENT: ANALYST IDENTIFIES SYSTEM CAPABILITIES FROM THE
C                         WEAPON/EQUIPMENT SYSTEMS FILE;
C                 RETRIEVE(D001.1,D001#1 EQ '$' AND D001#2 EQ '$');
C                 THINK(5,10,);
C                 COMMENT: THE ANALYST INDICATES THE RANGE ON MAP
C                          TO DETERMINE IMPACT ON CURRENT OPNS;
C                 CIRCLE(,RANGE = D001#40);
C           END-FUNC FIND-SYS-CAP.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: SEARCH-EVENTS           $$SRCHEV.ORA$$
C
C
C      ABSTRACT:
C              BEG-FUNC SEARCH-EVENTS;
C                 QUERY(B001.1,B001#23 EQ E001#2 AND B001#11 EQ '$');
C                 THINK(30,300,A);
C                 PAGE;
C                 THINK(30,300,A);
C              END-FUNC SEARCH-EVENTS.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C      PROCEDURE NAME: QUERY-INDICATORS        $$QRYIND.ORA$$
C
C
C      ABSTRACT:
C              BEG-FUNC QUERY-INDICATORS;
C                 COMMENT: THE ANALYST REQUESTS ALL INDICATORS
C                          FOR COUNTRY AND QUICKLY SCANS THEM;
C                 QUERY(F001.1,F001#1 EQ E001#2);
C                 THINK(30,150,A);
C                 PAGE;
C                 THINK(30,150,A);
C              END-FUNC QUERY-INDICATORS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ACTIVATE-INDICATORS      **ACTIND.ORA**
C
C
C       ABSTRACT:
C
C            BEG-FUNC ACTIVATE-INDICATORS;
C                 COMMENT: UPDATE THE INDICATOR STATUS;
C                 MODIFY(F001#24 = 'A');
C                 COMMENT: THE ANALYST MAKES HARDCOPY FOR LATER REFERENCES;
C                 HARDCOPY;
C                 THINK(5,10,);
C               END-FUNC ACTIVATE-INDICATORS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: NOTIFY-OPNS              ***NFYOPN.FTN***
C
C
C       ABSTRACT:
C               BEG-FUNC NOTIFY-OPNS;
C                  INSERT(, ,BLOCK = 'WATCH THIS AREA');
C                  THINK(2,5, );
C                  CHATTER OPNS-OFFICER;
C               END-FUNC NOTIFY-OPNS.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: BRIEF-ASSESSMENT         ***BRFASM.FTN***
C
C
C       ABSTRACT:
C               BEG-FUNC BRIEF-ASSESSMENT;
C                  THINK(300,600,A);
C                  AMP;
C                  THINK(300,600,A);
C               END-FUNC BRIEF-ASSESSMENT.
C
C       CALLED BY:      MAC-ROUTE-ASSESSMENT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: DISPLAY                 $$DISPLY.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'DISPLAY' COMMAND IS USED TO DISPLAY AN ALERT
C       QUEUE ENTRY TO THE TERMINAL.   THE RTE FORMAT IS:
C
C       DISPLAY [<alert number>][INTO FORM <userview>];
C
C       BOTH ARE OPTIONAL PARAMETERS, WHERE:
C       <alert number> - THE NUMBER OF THE ALERT QUEUE ENTRY TO BE
C                        DISPLAYED.  IF NOT SPECIFIED, DEFAULT TO
C                        HIGHEST PRIORITY ALERT QUEUE ENTRY.
C       INTO FORM <userview> - PARAMETER USED BY THE UPDATE SCENARIO
C                        TO DISPLAY AN ALERT QUEUE ENTRY INTO THE
C                        SPECIFIED <userview> FORMAT.
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: MAP                         $$MAP.FTN$$
C
C
C       THE DHIL 'MAP' COMMAND IS USED TO GENERATE A MAP BACKGROUND.
C
C       THE RTE COMMAND FORMAT IS:
C       MAP(COORD = <geo.coords.> | LOC = <location name>,
C            [SCALE = <scale>]);
C
C       PARAMETERS ARE:
C       COORD|LOC - MUST SPECIFY EITHER COORD OR LOC (BUT NOT BOTH)
C                   TO IDENTIFY THE CENTER POINT OF THE DISPLAY
C
C       SCALE     - OPTIONAL PARAMETER WHICH SPECIFIES THE SCALE OF THE
C                   MAP TO BE DISPLAYED.  SCALES AVAILABLE ARE: 1:20M,
C                   1:5M, 1:1M, 1:500K, 1:250K, 1:100K, 1:50K, 1:25K,
C                   1:10K, 1:7.5K.  IF NOT SPECIFIED, DEFAULT TO 1:20M
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PREPARE-MSG             $$PREMSG.ORA$$
C
C
C       ABSTRACT:
C               BEG-FUNC PREPARE-MSG;
C                 COMMENT: ANALYST RETRIEVES MESSAGE FORMAT AND ENTERS
C                          APPROPRIATE DATA;
C                 FORM A001.1;
C                 THINK(5,10, );
C                 ENTER(A001#01 = '$',A001#71 = 'I',A001#190 = 'SAM WARNING',
C                       A001#200 = 'MESSAGE TEXT');
C                 THINK(30,60, );
C                 COMMENT: THE WARNING MESSAGE WILL BE ADDED TO THE DATABASE
C                          MESSAGE FILE,THE ANALYST WILL MAKE A PERSONAL COPY
C                          THEN ROUTE THE MESSAGE TO THE SUPERVISORS ALERT QUEUE;
C                 ADDON A001.1;
C                 HARDCOPY;
C                 THINK(5,10, );
C                 ROUTE(LOGIN = 'DIV-CHIEF',MSG-ID = A001#1,PRI = 'I',
C                       SUBJ = 'SAM WARNING');
C               END-FUNC PREPARE-MSG.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ALERT
C
C
C       ABSTRACT:
C
C          ALERT will simulate an analyst displaying numbers and types
C       of messages on the alert queue.  The actual messages are not relevant.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: THINK
C
C
C       ABSTRACT:
C
C          THINK simulates an analyst mulling over a decision.  Because of
C       this, the result is divided by the alert status.  Obviously, the
C       analyst will think less time in a crisis or war situation.
C
C
C       INPUT:          MIN     - INTEGER;  Minimum peacetime wait
C                       MAX     - INTEGER;  Maximum wait at any time
C                       ALERT   - INTEGER;  Alert status (1,2,3)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SCAN
C
C
C       ABSTRACT:
C
C          SCAN simulates an analyst scanning through the list of alert
C       messages.
C
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: PLOT                        **PLOT.FTN**
C
C
C       ABSTRACT:
C
C       THE DHIL 'PLOT' COMMAND IS USED TO DISPLAY DBMS ITEMS ONTO A
C       PREVIOUSLY GENERATED MAP BACKGROUND.
C
C       THE RTE COMMAND FORMAT IS:
C       PLOT([<userview>,<fieldname> <relop> <fieldvalue> [<logop>
C             <fieldname> <relop> <fieldvalue>]],CLASS = <symbol>,
C             [BLOCK|NOBLOCK]);
C
C       PARAMETERS ARE:
C       <userview>,<fieldname>...<fieldvalue> - OPTIONAL SELECTION LIST
C               WHICH WHEN SPECIFIED CAUSES A DATABASE ACCESS FOR ALL RECORDS
C               THAT MEET THE CRITERIA TO BE PLOTTED ONTO THE MAP.
C               IF NOT SPECIFIED, PLOT THE INFO. ASSOCIATED WITH THE RECORD
C               CURRENTLY BEING DISPLAYED.
C       CLASS - REQUIRED PARAMETER WHICH INDICATES THE CLASSIFICATION OF
C               SYMBOLIC DATA TO BE PRESENTED ON THE MAP
C       BLOCK|NOBLOCK - OPTIONAL PARAMETER WHICH SPECIFIES WHETHER A 'DATA
C               BLOCK/TAG' IS TO BE APPENDED TO EACH SYMBOL GENERATED FOR
C               DISPLAY.  BLOCK IS THE DEFAULT.
C
C       NOTE: THIS IS THE ONLY DHIL COMMAND THAT WILL EITHER BE EXECUTED AS
C             A 'WAIT' TIME (WHEN ISSUED WITH NO SELECTION LIST), OR MAKE A
C             DATABASE ACCESS (WHEN ISSUED WITH A SELECTION LIST).
C
C       *********************************************************************
C       *********************************************************************
C       ****THIS 'PLOT' FUNCTION SHOULD ONLY BE CALLED BY FUNCTIONS WHICH****
C       ****ARE ISSUING A DHIL PLOT COMMAND WITHOUT A SELECTION LIST*********
C       *********************************************************************
C       *********************************************************************
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: INSERT                  **INSERT.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'INSERT' COMMAND IS USED TO ADD A NEW ITEM TO THE
C       SPECIFIED LOCATION ON THE CURRENT MAP DISPLAY WITH A DATA BLOCK
C       APPENDED TO IT, OR TO ASSOCIATE A DATA BLOCK WITH AN ITEM ALREADY
C       ON THE CURRENT MAP DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       INSERT([CLASS = <symbol>],[LOC = <value>],BLOCK = <value>)!
C
C       PARAMETERS ARE:
C       CLASS = <symbol> - OPTIONAL PARAMETER WHICH SPECIFIES THE SYMBOLOGY
C                          CLASSIFICATION OF THE ITEM BEING ADDED TO THE
C                          DISPLAY.
C       LOC = <value>    - OPTIONAL PARAMETER WHICH SPECIFIES THE LOCATION
C                          VALUE OF THE ITEM BEING ADDED TO THE DISPLAY.
C                          IF NOT GIVEN, DEFAULT TO THE CURRENT LOCATION
C                          OF THE CURSOR/LIGHT PEN.
C       BLOCK = <value>  - REQUIRED PARAMETER WHICH SPECIFIES THE DATA BLOCK
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: SEG                     **SEG.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'SEG' COMMAND IS USED TO DRAW A LINE SEGMENT
C       BETWEEN TWO POINTS ON A DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       SEG([FROM-LAT = <from-latitude>, FROM-LON = <from-longitude>],
C           TO-LAT = <to-latitude>, TO-LON = <to-longitude>)!
C
C       PARAMETERS ARE:
C            FROM-LAT AND FROM-LON ARE OPTIONAL PARAMETERS WHICH SPECIFY
C       THE STARTING POINT OF THE SEGMENT TO BE DRAWN. IF NOT SPECIFIED,
C       DEFAULT TO THE CURRENT POSITION OF THE CURSOR/LIGHT PEN.
C            TO-LAT AND TO-LON ARE REQUIRED PARAMETERS WHICH SPECIFY THE
C       ENDING POINT OF THE SEGMENT TO BE DRAWN.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CIRCLE                    **CIRCLE.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'CIRCLE' COMMAND IS USED TO DRAW A CIRCLE WITH THE
C       SPECIFIED RANGE AROUND AN ITEM THAT HAS BEEN IDENTIFIED ON THE
C       CURRENT MAP DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       CIRCLE([<fieldname><relop><value>|BLOCK<relop><value>],
C              RANGE = <range>);
C
C       PARAMETERS ARE:
C       <fieldname><relop><value> - OPTIONAL PARAMETER WHICH SPECIFIES THE
C              ITEM ON DISPLAY WHICH SHOULD BE CIRCLED.
C       BLOCK<relop><value> - OPTIONAL PARAMETER WHICH IDENTIFIES THE ITEM
C              WHICH SHOULD BE CIRCLED THROUGH IT'S DATA BLOCK VALUE.
C       RANGE = <range> - REQUIRED PARAMETER WHICH SPECIFIES THE RADIUS OF
C              THE CIRCLE TO BE DRAWN. (VALID UNITS ARE NM AND KM, DEFAULT
C              TO KM IF NOT SPECIFIED)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: HARDCOPY                  **HRDCPY.FTN**
C
C
C       ABSTRACT:
C
C            THE DHIL 'HARDCOPY' COMMAND IS USED TO FORMAT AND TRANSFER
C       ALL OF THE CURRENT DISPLAY TO A HARDCOPY DEVICE.
C
C       THE RTE COMMAND FORMAT IS:
C            HARDCOPY;
C       THE COMMAND WILL BE ISSUED AS A WAIT TIME FOR THE BENCHMARK
C       EXERCISES.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: INSERT                $$INSERT.FTN$$
C
C       ABSTRACT:
C
C            THE DHIL 'INSERT' COMMAND IS USED TO ADD A NEW ITEM TO THE
C       SPECIFIED LOCATION ON THE CURRENT MAP DISPLAY WITH A DATA BLOCK
C       APPENDED TO IT, OR TO ASSOCIATE A DATA BLOCK WITH AN ITEM ALREADY
C       ON THE CURRENT MAP DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       INSERT([CLASS = <symbol>],[LOC = <value>],BLOCK = <value>)$
C
C       PARAMETERS ARE:
C       CLASS = <symbol> - OPTIONAL PARAMETER WHICH SPECIFIES THE SYMBOLOGY
C                          CLASSIFICATION OF THE ITEM BEING ADDED TO THE
C                          DISPLAY.
C       LOC = <value>    - OPTIONAL PARAMETER WHICH SPECIFIES THE LOCATION
C                          VALUE OF THE ITEM BEING ADDED TO THE DISPLAY.
C                          IF NOT GIVEN, DEFAULT TO THE CURRENT LOCATION
C                          OF THE CURSOR/LIGHT PEN.
C       BLOCK = <value>  - REQUIRED PARAMETER WHICH SPECIFIES THE DATA BLOCK
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: CHATTER               $$CHTTER.FTN$$
C
C       IDENT:          680-007.XX
C
C       TASK NAME:      ANY SCRIPT OR FUNCTION, LINK W/DHIL.OLB
C
C       AUTHOR:         , Mc2
C
C       ABSTRACT:
C
C            THE DHIL 'CHATTER' COMMAND IS USED TO SEND A COPY OF THE
C       CURRENT TERMINAL DISPLAY TO ANOTHER ANALYST THROUGH THE ALERT
C       QUEUE.
C
C       THE RTE COMMAND FORMAT IS:
C       CHATTER <receiving LOGIN identifier> [,<receiving LOGIN identifier>]$
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: AMP                        $$AMP.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'AMP' COMMAND IS USED TO DISPLAY TEXT DATA
C       (IN A RESERVED AREA OF THE SCREEN) ASSOCIATED WITH THE SPECIFIED
C       ITEM/LOCATION ON DISPLAY.
C
C       THE RTE COMMAND FORMAT IS:
C       AMP([BLOCK <relop> <value> | LOC <relop> <value>]);
C
C       PARAMETERS ARE:
C       BLOCK - VALUE IN DATA BLOCK ASSOCIATED WITH A SYMBOL
C       LOC   - LOCATION OF A DATABASE ITEM (i.e.; coordinates)
C            'AMP();'  CAUSES AMPLIFICATION OF ITEM IDENTIFIED BY
C       CURRENT CURSOR/LIGHT PEN POSITION.
C
C       ACTUAL IMPLEMENTATION WOULD CAUSE THE RETRIEVAL OF BLOCK INFORMATION
C       FROM A DATABASE OR EXTERNAL FILE.  THE SCHEMA DEFINITION HAS NOT
C       INCLUDED THIS POSSIBILITY, AND THERE ISN'T MUCH SPACE AVAILABLE...
C       SO, HANDLE AS A 'WAIT' TIME.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: FORM                     $$FORM.FTN$$
C
C
C       ABSTRACT:
C
C            The FORM command is used to display the template
C       format for the specified userview.
C
C       The RTE command format is:
C            FORM <userview>;
C       This command is used to display a form which will then
C       have information ENTERed into it by the analyst (a sequence
C       of steps used to add a record to the database).  For testing
C       purposes this command will be executed as a WAIT time.
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ENTER                    $$ENTER.FTN$$
C
C
C       ABSTRACT:
C
C            THE DHIL 'ENTER' COMMAND IS USED TO SIMULATE THE ANALYST
C       ENTERING DATA INTO A PREVIOUSLY DISPLAYED FORM.  ONLY THE CONTENTS
C       OF THE DISPLAY ARE MODIFIED -- THIS COMMAND HAS NO EFFECT ON THE
C       DATABASE.  SUBSEQUENT PRIMITIVE DIRECTIVES CONTROL THE DISPOSITION
C       OF THE DATA.
C
C       THE RTE COMMAND FORMAT IS:
C       ENTER(<fieldname> = <new value>[,<fieldname> = <new value>]);
C
C       PARAMETERS ARE:
C            <fieldname> - THE NAME OF THE FIELD RECEIVING DATA
C            <new value> - VALUE TO BE ASSIGNED TO RECEIVING FIELD
C
C       FOR THE RTE, THIS COMMAND WILL BE EXECUTED AS A 'WAIT' TIME
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C       PROCEDURE NAME: ROUTE                    $$ROUTE.FTN$$
C
C
C       ABSTRACT:
C
C            The DHIL command ROUTE is used to insert message information
C       into the alert queue of an analyst.
C
C       The RTE command format is:
C       ROUTE(LOGIN = <receiving analyst>, MSG-ID = <field value>,
C             PRI = <field value>, SUBJ = <field value>);
C
C       Required parameters are:
C       LOGIN  -- the login identifier of the receiving analyst
C       MSG-ID -- the msg-id-code which uniquely identifies a message
C                 within the databas
C       PRI    -- the message priority
C       SUBJ   -- brief information indicating the subject/content of
C                 the message
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

APPENDIX D. TASK INFORMATION BLOCKS (TIBS)

This appendix provides a more detailed description of the the Task Information Blocks (TIBs) used in generating the I&W data base. They are important in showing what the detailed design of an I&W data base might look like. While specific formats are somewhat arbitrary, the design gives a picture of an overall data base which can support the I&W function as defined by the scenarios.

Because of the amount of information contained in each TIB, only one of them is shown in detail, in Table D-1. Other TIBs are more briefly described in the text that follows. A full set of TIBs is contained in the Interim Technical Report for this project.

TIB #A001, Message Processing Input/Output, is based on the format for GENSER message traffic. It can be accessed only through its unique ID code. The format for A001 is illustrated in Table A-1.

TIB #B001, Event Log, contains extensive information concerning significant events. The type of event and the actors in the event, together with estimates of source reliability and accuracy, are included. Time and place may be used as search keys, as well as the organizations participating in the event. Detailed identifications of personalities participating in the event, weapons employed, and an event chronology are included. Other associated events are cross-referenced.

TIB #C001, Order of Battle, includes standard forms for OB data. The participating unit is identified, unit organization is described, and information concerning location is included. Other optional information includes unit movement information, personnel, equipment, weapon systems, and tactics. Information concerning logistics, combat effectiveness, and related personalities is also included.

TIB#: A001   Functional Area:

TIB Name: Message Processing Input/Output

Purpose/Use: Based on the format described for GENSER message traffic

Related TIBs:

Access Rights:

Item	Item	Size/Access		
No.	Description	Struct.	Use	Remarks
+01	Header			
1.	Incoming message unique ID code	15AN	Entry	
+02	Format line 1			*All files in Block 1 should be packed (no spaces) for display in 69 characters or less on one line.
21.	Validation character	1A		
22.	Status of message indicator	4A		
23.	Start of message indicator	3A		
24.	Channel sequence number	3N		
+03	Format line 2			
31.	Communications routing information	69AN		*Usually not shown.
+04	Format line 3			
41.	Prosign	2A + 1 blank		*Display as

Table D-1 Examples of Task Information Block (TIB) (Page 1 of 4)

Item No.	Item Description	Size/Access Struct. Use	Remarks
42.	Originator station routing indicator	7A + 1 blank	one line as indicated with blanks -- not to exceed 69X characters total line.
43.	Station serial number (#____)	5X + 1 blank	
44.	Julian date	3N	
45.	Zulu time	4N	
+05	Format line 4		*+05 and +06 refer to Format line 4 -- consisting of one or more lines of not more than 69 printed chars.
51.	Security warning operating signal	3A	
52.	First letter of message security classification	5A	
53.	Communications operating signal	23X	
+06	Tango addressee		*May repeat 100+ times. Each addressee separate line.
61.	Special transmission	23X followed by "T"	
+07	Format line 5		
71.	Action precedence	1A + 1 blank	*Line 5 may contain communications operating signals -- both of which can repeat -- but the entire line cannot exceed 69X.
72.	Info. precedence	1A + 1 blank	
73.	Date-time group of transmission	15AN	
+08	Communications operating signals		
81.	Signal for book message	3A - optional	
82.	Signal for correction	3A - optional	

Table D-1 Examples of Task Information Block (TIB) (Page 2 of 4)

Item No.	Item Description	Size/Access Struct. Use	Remarks
+09	Format line 6		*Line is limited to 58X
91.	FM	2A + 1 blank	
92.	Originator's name and address	55X	*Begin line with "FM "
+10	Format line 7		*The symbol "TO" appears first line only.
100.	TO	2A + 1 blank	
+11	Addressee		
110.	Routing indicator/ official name	55X	*Addressee may repeat 500+ times.
+12	Format line 8		
120.	INFO	4A + 1 blank	*Line 8 begins with "INFO " followed by addressees -- same conventions as line 7.
+13	Addressee		
130.	Routing indicator/ official name	55X	
+14	Format line 9		*Same as lines 7 and 8 -- first line begins with "XMT "
140.	XMT	3A + blank	
+15	Addressee	55X	
150.	Routing indicator/ official name		
+16	Format line 10		*Not used

Table D-1 Examples of Task Information Block (TIB) (Page 3 of 4)

Item No.	Item Description	Size/Access Struct. Use	Remarks
+17	Format line 11		*Line 11 consists of text-boundary symbol "BT" only.
170.	BT	2A	
+18	Format line 12		*Spaces between letters except for UNCLAS. *Not strictly formatted.
180.	Security classification		
181.	Handling caveats		
182.	Sectional message ident.		
+19	Subject line		
190.	Subject line	69X	
+20	Text		*May repeat 5000+ times.
200.	Message text	69X	
+21	Format line 13		*Contains text-boundary symbol only.
210.	BT	2A	
+22	Format line 14		*Not used.
+23	Format line 15		
230.	Station serial number	5X	*Same as +04 #43
231.	Two carriage returns	2X	
232.	Eight linefeeds	8X	
+24	Format line 16		
240.	NNNN	4A	*Contains only "NNNN".

Table D-1 Example of Task Information Block (TIB) (Page 4 of 4)

TIB #D001, Weapons/Equipment Summary, provides formats for storage of information concerning weapons of all types. Technical characteristics, such as dimensions, weights, velocity, and range, are stored and may be used for retrievals.

TIB #D002, Personalities, stores names, aliases, and detailed descriptions of persons of interest. This TIB also includes information concerning the source and reliability of each descriptive item.

TIB #E001, Country Summary Information, contains information concerning individual countries. Estimates of the stability and the status of the current government are among the items included. International affiliations and allegiances, and the general level of hostility, are indicated. Data described by this TIB would be used in route determination or in evacuation planning, where the degree of hostility of each area will have an effect on the plan.

TIB #E002, Installation Information, includes descriptions of various types of installations. In addition to locations, the files described by this TIB will include activity in the following categories: aircraft/vehicle, construction, weapons, shipping/transportation, industrial, road, rail, and other categories. Defenses, runways, hangars, dispersals, weapon storage, fuel storage, electronics, access, security, and unusual features are also referenced. Primary and secondary units/structures are described. Dimensions of the installation, general and specialized facilities, antennas, and power sources are also included for some installations.

TIB #F001, Indicator Lists, contains information concerning current indicator status. The country to which the indicator pertains, together with a classification of the indicator, is also included for use in retrieval.

TIB #G001, Collection Request, represents the standard joint Tactical Air Reconnaissance/Surveillance Request Form used by the analyst. In addition, information used by the Collection Coordinator is also included.

TIB #H001, Friendly Mission Information, specifies information concerning type of mission, date and time, location, destination, expected arrival date and time, and details concerning the mission units and aircraft.

TIB #I001, Air Route Description, provides a general description of the route, including the type of mission generally using the route, and indicates weather conditions and threats on the route. Route restrictions are also described.

TIB #I002, Air Route Segment Description, provides more detailed information concerning individual route segments. Descriptions and locations of individual segments, details of threats, facilities located on the route segment, and features and obstacles are listed. Navigation aids are presented in detail. An assessment of the suitability of the segment for various mission types is included.

TIB #J001, Weather Summary, describes current weather conditions in selected localities. Information concerning general weather conditions is included, and provision is made for severe weather warnings.

Appendix E. GENERAL PURPOSE MONITOR

The GPM was designed to operate on DEC PDP-11/70 equipment under the IAS Operating System. The system configuration is shown in Figure E-1.

The User Interface consists of a set of indirect command files, which the user must execute in order to provide input. These inputs are used to produce data buffers for use by the monitor. A series of indirect command files can be used to specify different monitoring options, as shown in Figure E-2:

- o MCCNMS - task names
- o MCCXRV - send/receive directives
- o MCCQIO - QIOs
- o MCCFLS - filenames for monitoring I/O
- o MCCTEX - task execution directives
- o MCCTIN - miscellaneous directives
- o MCCSEW - exit, stop, and suspend directives

The user specifies the monitoring options for a monitoring session by executing the appropriate indirect command files. A monitoring session is then initialized and monitoring itself begins. At the end of the session, a separate reporting task (MCCOUT) accesses the disk resident monitored statistics to provide formatted reports on disk statistics, node use, task statistics, directive use, and I/O to file names.

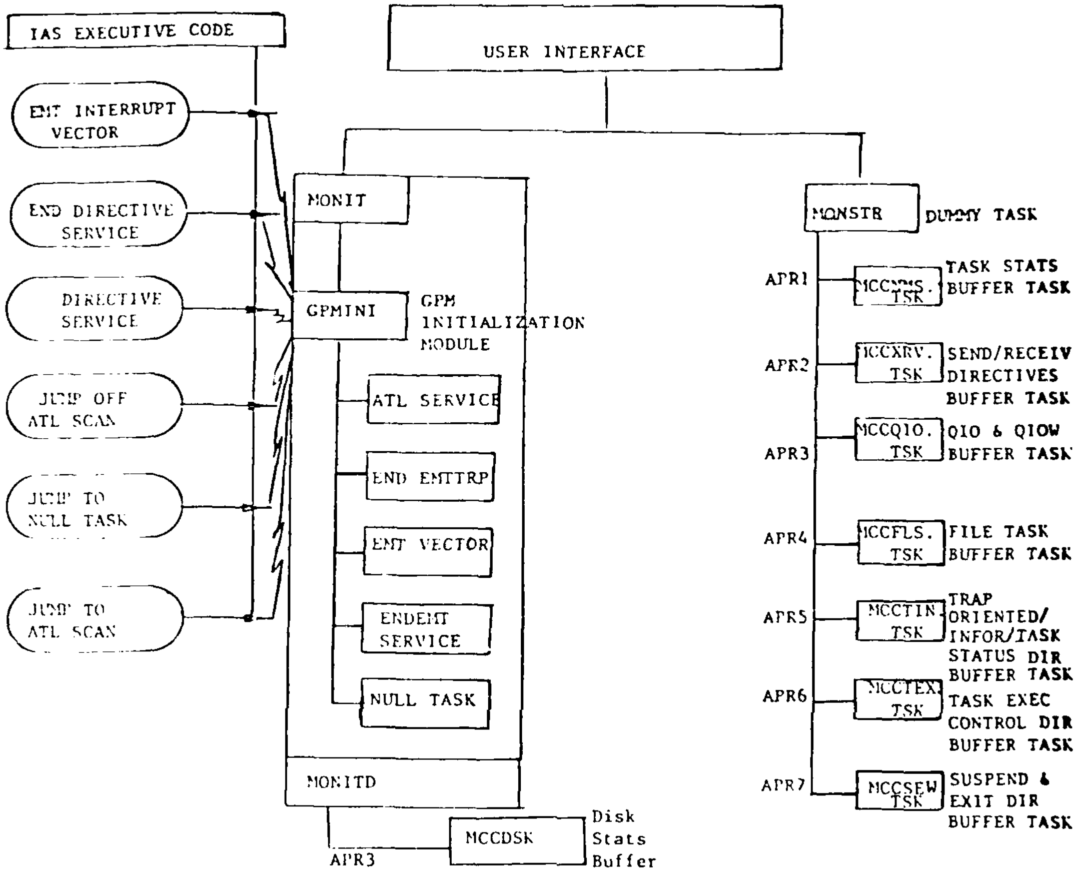
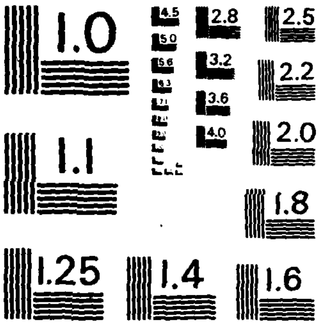


Figure E-1 Monitor Configuration

AD-A139 339 I&W (INDICATIONS AND WARNING) DATA BASE MANAGEMENT 4/4
SYSTEMS ANALYSIS(U) MEASUREMENT CONCEPT CORP ROME NY
JUN 83 RADC-TR-83-136 F30602-80-C-0286

UNCLASSIFIED F/G 5/2 NL


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

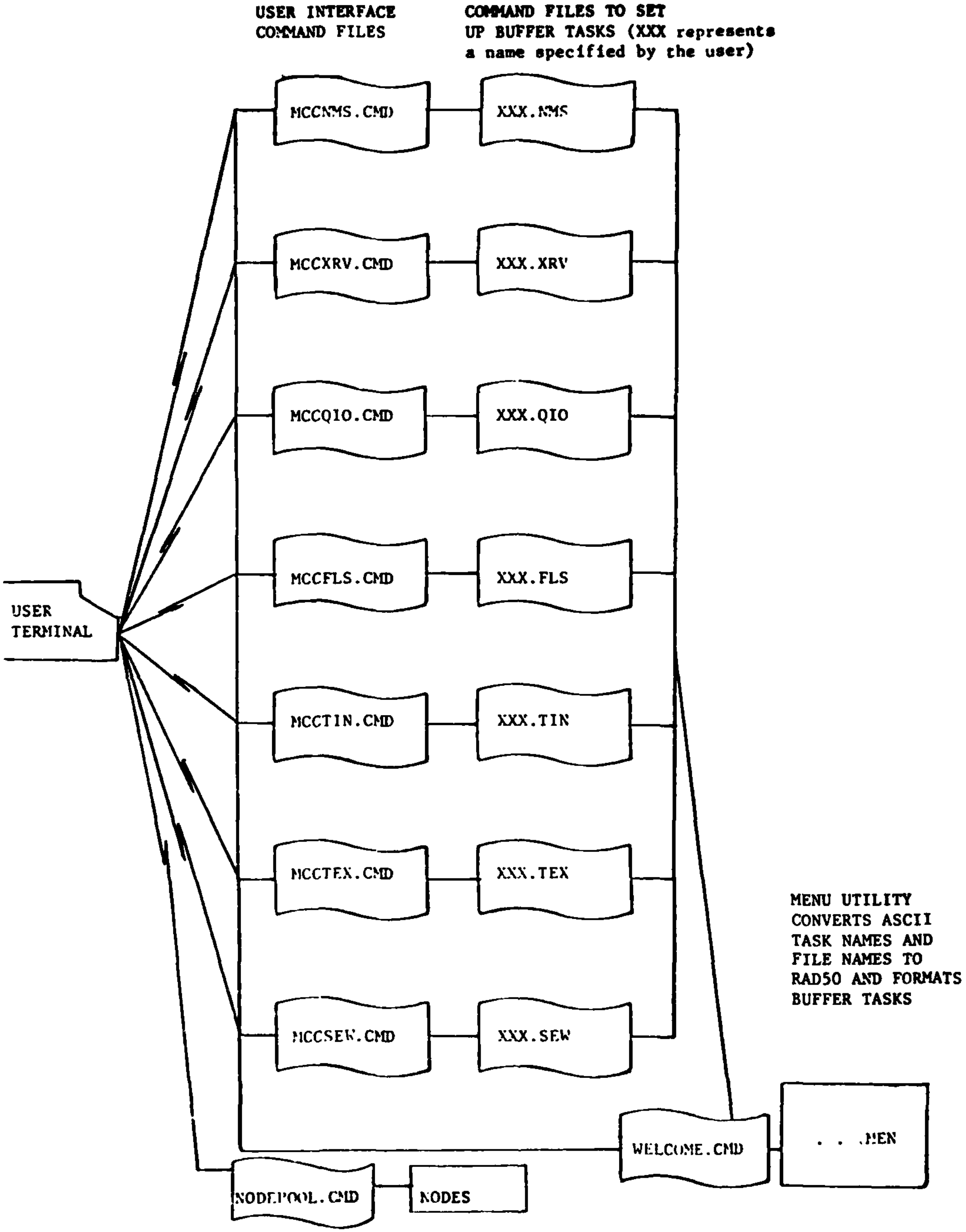


Figure E-2 User Interface

The processing time, amount of core memory use, and number of files output by the GPM are a function of the number of options selected by the user. Any combination of options may be selected. The GPM creates a set of statistical files for each monitoring session that is conducted. Each individual file generated contains the results of monitoring only certain activities: a group of EMTs, certain task names, particular disk drives, etc. Subsequent monitoring sessions will generate a different set of files, thus allowing the user to save historical statistics as long as they are required.

During the Remote Terminal Emulation (RTE) tests, the GPM was used to obtain statistics required for evaluation of the three systems under test. The results of these tests are included in Section 12.0 of this report.

Appendix F. FURTHER RESEARCH

F.1 Artificial Intelligence Experiments

This section describes potential further use of the test facility developed during this effort. The illustrations provided here are intended to show the manner in which the scenarios, data bases, and other components may be applied to the evaluation and demonstration of other systems and approaches. Techniques to be defined and described here are termed "Artificial Intelligence" (AI) in the sense that they replicate, emulate, or augment human cognitive processes.

The following examples illustrate AI techniques and potential I&W applications:

- Pattern recognition methods may be used to identify recurring patterns of events, for the purpose of identifying significant events or predicting future events. Correlations among messages could be used to form clusters of messages referencing the same or related events.

- Automatic update facilities could be provided, in which incoming free-text messages are analyzed to provide formatted information for update of data bases.

- Techniques for language translation may be used to provide natural language query facilities for a data base system.

- Methods developed in an experimental context for question-answering programs could be adapted to extend the capabilities of a data base system.

o Programs for logical inference may be employed to show the implications of current events.

o Decision analysis systems could be used to assist the intelligence officer in supplying evidence for a warning or other projection.

o A knowledge-based system, containing information or procedures derived from the practice of experienced intelligence analysts, could be developed.

The tasks described here are intended to determine whether AI techniques can support I&W Intelligence analysis, using data bases, scenarios, and other software implemented for this effort.

Several types of AI have already been studied intensively for application to the I&W environment. They include:

- The Advanced Indications Technology Experiment (AITE) and its successor, the Prototype Advanced Indications System.
- Waveform recognition techniques applied to identification of radar and other electronic emissions.
- Image recognition methods used for analysis of aerial photographs.
- Automatic translation of foreign language information.
- Analysis of messages for automatic dissemination.

The techniques and experiments represented by these developments are significant applications of AI techniques to I&W information requirements, illustrating the practical importance of AI research. The purpose of the experiments outlined here is to evaluate additional AI applications, using the I&W DBMS evaluation facility.

The field of AI is not well defined, but may be generally characterized as the application of techniques based on human information processing, rather than traditional mathematical, statistical, or computational approaches.

For example, in its early versions, AIS relied heavily on curve fitting and other traditional mathematical approaches to obtain predictors of significant events. The number of events, such as long-range aircraft flights, was tallied, and significant variations from the norms were noted. These provided indicators of potentially signficant changes. This early approach in AIS was a non-AI, traditional mathematical solution.

Later versions of AIS more closely resemble AI, since they use reasoning processes more like those of human experts. Human reasoning typically takes the form of a conditional, if-then statement: If the Soviets launch a satellite from Base X on a Sunday, rather than its normal weekday launch, then they have a special interest in the area under surveillance by the satellite. Like a knowledge-based AI system, recent experimental versions of AIS contain a large number of rules of this type. Typical rule-oriented systems resemble human experts in relying on somewhat informal heuristics, or rules of thumb, rather than on formal mathematical methods of data analysis and reduction.

For the purpose of the tests described here, we will use a portion of the EWAMS/WEIS data base (see Section 6.0). Another data base for consideration would be the Libyan scenario material, using Foreign Broadcast Information Service (FBIS) sources (see Section 7.0).

F.1.1 Pattern Recognition

Pattern recognition techniques are available in two general forms: statistical and syntactic. Statistical methods include the use of the Fisher discriminant, Bayesian classifiers (and more tractable substitutes for them), factor analysis and other statistical clustering techniques, various regression methods, and other statistical approaches. Graphic methods are sometimes combined with statistical methods, to permit the experimenter to visualize the structure of the data, and to draw lines between clusters of data items that appear to be distinct.

Syntactic methods are somewhat newer and more closely related to other AI approaches. Essentially, they use production rules, resembling syntactic rules in linguistic analysis or automata theory, to describe the requirements for inclusion in a pattern class. A syntactic pattern recognition system thus is similar to a rule-based AI system. [Cf. Paul R. Cohen and Edward A. Feigenbaum, Handbook of Artificial Intelligence, Los Altos: William Kaufmann, 1982, Vol. III, pp. 287-291.]

Pattern recognition and correlation techniques may be used to recognize and classify events. An event is a change in the values associated with an attribute of an entity. A significant event is one that requires a response, such as further study, or issuance of a warning or other message. Event analysis includes:

- Determination of the classification to which an event belongs.

- Clustering of messages which refer to the same or similar events.

- Identification of common patterns or family resemblances among related events.

- Prediction of future events.

The goal of event analysis is the development of a coherent picture or model of the event. The model might take the form of a set of statements relating the various components of the event into a whole. To make sense of the event, the model would include an assessment of enemy intentions and capabilities as related to further events. The model thus serves as the basis for predictions and for warnings.

The following components of event analysis were listed in the Benchmark Test Plan:

- o Event correlation: locating other data records which reference similar events, or other aspects of the same event.

- o Event verification: locating other records which will tend to increase or decrease the estimated reliability of the report.

- o Event expansion: locating other records which contain additional data concerning causes or effects of the event.

- o Event coordination: locating other records of related or similar events.

The Benchmark Test Plan for this project stated that it was possible to perform these operations, forming a composite or coordinated event record in the analyst file, which contains references back to the supporting data. It should be possible for the analyst to prepare a data record for the hypothesized event and report concerning the event, using report generation facilities. It should also be possible to display the information against a cartographic background.

The first method for grouping messages will be to use human visual capabilities to identify clusters of geographically related items. Development of demonstration facilities for the I&W DBMS Analysis has made it possible to display locations of elements of the WEIS data base against a cartographic background. A map of the selected area is generated at the color graphics terminal, and message locations are displayed as colored dots on the map. Using a cursor, the user can point to a dot on the map, which will cause the corresponding message to be displayed on another terminal.

The next step will be to permit the user to draw a boundary around a set of messages, and to retrieve all messages lying within this area. As the message locations are displayed, the user will be able to identify clusters of messages that represent a single event, or a group of closely related events.

Since the demonstration software permits an area search, this facility would not require any extensive further software, but it would require some method for determining whether a message was or was not within the boundary that the user has drawn. The simplest approach would be to require the user to draw the boundaries along coordinate lines and use existing software to locate messages within this area.

A second method for grouping messages is to identify time clusters, i.e. sequences of messages within a limited time period. Since time alone is not enough to identify messages with related subject matter, this method may be used in conjunction with geographic clustering. The following experiment would show the usefulness of this approach:

The user first identifies a geographic area of interest, and requests the system to store all messages from within this area. Next, the messages are sorted by time of occurrence (if they are not already in time sequence). The system then displays the locations of the messages in sequence, from earliest to latest. This time-sequence display should give a clear picture of the development and geographical structure of an event, permitting the analyst to extrapolate to future developments. The geographical pattern of the event can be identified.

Another approach would identify patterns within the event descriptions themselves, and could serve as the basis for locating related or similar messages. The techniques described here have been used extensively for content analysis of free-text data bases at Cornell and Syracuse Universities.

The goal of content analysis is to identify clusters of related messages (or other texts) within the data base, which may be presumed to deal with the same or closely related topics. In general, an ideal cluster is a set of items which are more highly correlated to other items in the cluster than to any items outside the cluster. Since real clusters are seldom ideal, clustering techniques use various compromises to obtain useful groups of items. Each member of a group or cluster of correlated messages is an example of the pattern that identifies members of the cluster. This pattern can be used as the key to recognizing other potential members of the group.

The following steps are usually performed in developing a content analysis system of this type:

o Review data base to remove non-content words (and, or, is, but, etc.) and to remove syntactic word endings (-ed, -ing, -s, -ment, -er, etc.), leaving word-stems.

o Identify and discard word-stems which will not be useful for clustering, such as words which occur only once (and thus do not correlate with anything) and words which occur in nearly all the messages (and thus would correlate with everything).

o For the k words remaining after this culling process, form a k x k correlation matrix, where the more highly correlated words are those that occur in the same messages with the same frequency.

- For the n messages, form an n x n correlation matrix, where the more highly correlated messages are those which contain the same words, or highly correlated words.

- Form clusters of highly correlated messages, representing the patterns that occur in the data base.

The process of cluster formation is a non-trivial task, for which packaged clustering routines are available. The message patterns represented by the clusters can now be used for automatic identification of other messages dealing with the same event ór subject area. A fully developed pattern recognition system might be used for the identification of significant events in the stream of incoming I&W messages.

A pattern recognition system of this type could be used to perform the four operations for event analysis described above: correlation, verification, expansion, and coordination. A substantial software effort is required to provide the needed correlation routines, but the existence of the I&W DBMS analysis software can provide much of the required support for this system.

In addition, it will be possible to obtain summary statistics from the WEIS/EWAMS data base, particularly from the free-text portions, which could be used to estimate the usefulness of this approach. Specifically, a table of words contained in the free-text portions of the WEIS data base should be prepared and reviewed to determine, on an intuitive basis, whether statistical correlation techniques might be valuable for developing a system that would recognize patterns in the data. This review should be complete enough to justify a recommendation concerning further work in this area.

A completely different approach to pattern recognition in the WEIS data base would use syntactic pattern recognition methods. This approach, which is similar to that of knowledge-based AI systems, is discussed in more detail in the last section of this report.

F.1.2 Automatic Data Base Update

Automatic data base update is regularly performed at ADCOM in processing formatted data, such as data concerning position and course of space objects. This information is extracted and inserted automatically in the Space and Missile OB files. The goal of this study is to determine whether automatic data base update is feasible when the source of information is unformatted or partially formatted data. For example, information concerning the location of Soviet submarines might be used to update the appropriate OB, without analyst intervention.

In fully developed form, such a facility would review incoming messages, such as GENSER traffic, identify the message subject on the basis of key words appearing in the text (GENSER messages do not always include a separate subject line), send the message to a specialized routine for data extraction, and, finally, issue a call to the DBMS to perform the actual updating procedure. Because of the critical importance of the data contained in intelligence files, redundancy and internal validity checks must be provided within the system to keep the error level within tolerable limits and to reduce the possibility of catastrophic error, which might include:

- Failure to identify update message.

- Misidentification of update message, with update performed on wrong files.

- Insertion of data into wrong fields, or data in wrong format.

- Cascading errors, i.e. errors in data items affecting other data items, or growing in magnitude with each update.

- Accidental destruction of existing data.

This list is not meant to be exhaustive, but simply to illustrate some of the opportunities for error in an automatic update system. Similar errors can occur in manual update systems, but there is no information to show that automatic updates are more or less error-prone than manual updates.

Within the context of the I&W DBMS Analysis facility, it is possible to demonstrate and test automatic update procedures, using existing data bases and the DBMSs under test. The general goal of this demonstration would be to determine the applicability of each DBMS to such procedures. GENSER traffic should be used if available; however, since there may be some delay in obtaining GENSER material, this discussion will assume that the WEIS/EWAMS data base will be used. This will simplify the problem, but it should also provide a demonstration that the approach is a reasonable one; conversely, if the problem of automatic update cannot be performed with the relatively simple WEIS data base as its source, it is likely to even more difficult to use GENSER or other major military data sources.

The first step in this demonstration is the use of routines to extract information from the formatted section of the WEIS data. Tables should be prepared showing the number of interactions between countries, the number of hostile vs. non-hostile interactions, and the number of interactions of each type.

Next, it should be possible to update tables automatically. A new WEIS message is read, and the formatted sections are translated into a call to the appropriate record in the data base. The existing value is read, the new data item is added to it, and the updated value is returned to the data base. For example, if a hostile interaction between Israel and Syria is input, the program will obtain the existing number of hostile interactions between these countries, add one to it, and return the new value to the data base.

The next step would be to review techniques for extracting data from free text materials. Here, the verbal sections of the WEIS data base could be used. They are considerably shorter than the GENSER messages and more stereotyped in format, but for that reason automatic translation would be more likely to succeed. The procedures would follow the same pattern as those suggested for the formatted portions of the WEIS data.

In this test, the problem would be to determine whether there is sufficient consistency in the form of the free-text summaries to permit automatic identification of hostile or cooperative interactions, and to identify the participating countries. Given the tables of hostile and non-hostile interactions described above, this function will attempt to provide automatic updates from the free-text portions of the WEIS data base.

Development of the automatic update mechanism for use with WEIS data must next be extrapolated into the substantive problem of providing an update mechanism with GENSER data. The question to be asked is whether this approach is likely to be effective when a realistic intelligence data source is used. GENSER messages should be reviewed to determine:

- o Which messages are likely to be used as inputs to an automatic update facility?
- o Of these messages, what fields or key words are present that could be used to identify and classify them?
- o Will it be possible to design an algorithm for extraction of required data from the messages?
- o Can the extracted data be used in a call to one of the DBMSs to update the appropriate data base?
- o What opportunities for error are there in this process, and how can the destructive effects of errors be prevented?

F.1.3 Query Facilities

In the original proposal for this project (Section 3.6.2) DHIL was described as potentially "the standard I&W data base access language." Subsequent clarification from RADC indicated that full development of a standard language was beyond the scope of the project, and DHIL was tailored specifically to the job of I&W scenario development. Later, the orginal project plan was modified, with the approval of RADC, to eliminate the projected use of automatic translators from the DHIL into individual data base access languages. The translations were performed manually. The task of translating DHIL statements into forms required by each of the DBMSs has provided a substantial fund of information concerning the requirements of a generalized data access language, and this information should be preserved as background for a design document outlining the requirements and possible formats of the language.

Should the DHIL be used as a prototype for a general purpose access language, as suggested in the project proposal? Should a natural language, such as English, be used as a lingua franca, a general language to access all data bases? Or should some alternative be chosen? Is it even possible to develop a generalized data access language? What can AI research contribute?

A substantial amount of AI experience has gone into developing systems for understanding natural language, and for answering questions posed in natural language. There is no substantial evidence, however, that natural language is the most effective medium of communication between humans and machines. The following are typical arguments against the use of natural language as a query language:

o Since reliable voice input is still not available, natural language must be entered through a typewriter keyboard, which most humans find difficult to use efficiently.

o Natural language is verbose, which means that a great deal of typing must be done, when a limited number of brief commands would be sufficient for most DBMS purposes.

o Queries which make good sense in vernacular English may be misinterpreted by automatic parsing algorithms.

o All existing natural language processors place severe restrictions on the vocabulary and syntax of the commands that they will recognize. These restrictions may be extremely frustrating to human users.

o Syntax and spelling errors may lead to misinterpretation or rejection of the command. Retyping an extensive command is burdensome for the user.

These objections are well known to proponents of natural language processing, and many recent systems take them into account. Nevertheless, it appears that natural language is a form of overkill for the design of an I&W DBMS front end:

o Since the DBMS performs a very limited set of functions, it is not necessary to provide the software required for processing a large subset of natural language.

o No query language permits the user to begin writing commands without any prior training. Since the user must be trained, it is possible that a simple artificial language will require less training time and effort, and will be more reliable in actual use, than a restricted subset of natural language.

o A simple query language could use mnemonics that are more easily remembered than natural language commands, given that the user is permitted to write only a limited portion of the full natural language. That is, the user could easily forget some of the restrictions on the natural language processor that he or she is using.

These objections suggest that for trained users, working with a limited set of objectives, the disadvantages of natural language may outweigh its immediate appeal. A specialized query language, in which the user needs to remember only a small number of commands, and which has a simple, straightforward syntax, may be more easily learned, may be more easily used, and will certainly be more easily implemented, than a natural language processor. DHIL could be considered as a prototype for such a language.

These doubts concerning natural language, however, should not be taken to be a rejection of all work that AI has completed in the development of question answering systems. On the contrary, some of the most important and interesting work that AI has done has been the analysis of user questions and the presuppositions that users bring to a system. [Belnap, Nuel D., Jr., and Steel, Thomas B. Jr., The Logic of Questions and Answers, New Haven:Yale, 1976; Lehnert, Wendy G., The Process of Question Answering, Hillsdale, N.J.: Lawrence Erlbaum Associates, 1978.] We should attempt to see how this AI research can contribute to the design of a user-oriented query language.

We have two possible goals for research in this area:

o Development of DHIL on the basis of experience in translation into DBMS languages. Development could mean scuttling DHIL as it currently exists and designing a new user language.

o Application of AI research in the logic of questions and answers, specifically in the problem of supplying information that better meets the user's needs.

In support of the first goa [illegible] hould be certain that we maintain adequate records of the ta [illegible] lating DHIL statements into the formats required by each of [illegible] DBMSs. Records should indicate difficulties in translation, useful additional commands, problems in syntax, ambiguities, and other information. The goal is to provide a fund of experience that can be used in designing a general purpose language for automatic translation.

In addition to its work in natural language processing, AI can contribute to the analysis of queries to provide more adequate responses. The following items are typical of the problems that AI considers:

- It should not be necessary to repeat a query in full, when it is closely similar to the preceding query. The analyst asks: "What is the displacement of the Krupny class destroyers?," and receives an answer from the system. Instead of repeating the full query, the analyst should be able to type simply: "Kanin?" with the system filling in the remaining information from memory.

- When the system fails to produce a hit, it should automatically broaden the area of the search until a hit is found. When performing a circle search, retrieving all items within a specified geographic area, the system should be able to expand the area when no hits are found, or when the analyst is not satisfied with the results.

- When searching for names, near-misses should be available, because of variations in spelling conventions. This facility is particularly important in searching for names which have been transcribed from interrogations and other verbal sources, where the transcribed spelling may vary from the standard.

These examples are well within the state of the art and, in fact, are implemented in some experimental data base systems. Within the scope of the I&W DBMS Analysis facility, the question will simply be whether advanced systems of this type will be available to users of the DBMSs under test. Questions like the following should be asked:

o Does the user have to specify all information in explicit form, or does the system make assumptions concerning the user's implicit information requirements?

o Does the system attempt to correct the user's spelling?

o If a syntax or spelling error is detected, does the user have to retype the entire query or command?

o Is there any sense in which the system learns from the user? For example, if the user always accesses the same file, will the system learn to open that file automatically? If the user frequently enters the same command, can that command be saved and reentered, or does the user have to retype it every time it is used?

o Does the DBMS permit abbreviated command syntax or spelling? Facilities like these have become so familiar that it seems exaggerated to call them "artificial intelligence." Nevertheless, they have grown out of AI research and are typical of practical applications of AI.

An AI system attempts to "understand" the user's query, which means that it identifies the presuppositions and implications of the query, going beyond what the user has specified explicitly. If the user asks, "How many Soviet fishing vessels are in the Bering Straits?," the system should be able not only to perform the required geographic search, but also to give more than a simple answer, "none," or "16" or "23." It should be able to say, "None, but there are 18 within 50 miles of the Straits," or "None, but there are 3 submarines in the Straits," or "None, because the Straits are frozen over." The system "understands" that the user is interested in identifying threats in the Bering Straits area, and it provides the user with enough information to meet this implicit goal.

F.1.4 Logical Inference

Theorem proving techniques have been sufficiently developed to permit the design of practical systems which depend on logical inference. Methods developed over the past twenty years extend far beyond the use of simple theorems to prove formulas in a logical calculus. For example:

o An approach to automatic computer programming uses theorem proving techniques to generate programs by "proving" assertions about the programs. The resulting proof is the required computer program.

o Instructions for the control of robots are generated by "proving" the result of their actions. The proof is a list of actions required to obtain the result.

In short, automatic theorem proving has practical applications, in addition to its theoretical applications in formal logic. (In fact, the practical applications may have greater long-term significance than the work in abstract logic, where one well-known result -- that there is no effective decision procedure for first-order predicate logics -- places an absolute theoretical limit on any theorem proving algorithm.)

Within the context of the I&W DBMS test facility, this research should focus on the question of meeting I&W analyst requirements in conjunction with the DBMSs under test:

o To what extent can theorem proving techniques improve the quality of the information that the I&W analyst can extract from an I&W data base?

Although modern theorem proving techniques have grown very complex, they tend to use an approach that we all learned in elementary logic classes: the use of truth tables. All possible combinations of truth values are generated to determine the conditions under which an assertion is true or false. If a set of assertions cannot all be true simultaneously (that is, if they "clash"), then one of them must be false. The goal of automatic theorem proving is to develop efficient methods for testing large numbers of complex assertions for clashes.

One application to I&W would be to test the assertions contained in a data base to locate anomalies or inconsistencies. This would require a consistent coding of WEIS data in a format that could be translated into the logical notation used in theorem proving systems. In an experimental system, the data could be reviewed to identify errors, anomalies, and changes over time.

Another potential application would use theorem proving techniques for locating significant events, such as threats. The threat would be entered as the conclusion of a logical deduction, and the system would be required to prove that the threat exists. The proof sequence would provide the evidence required in the analyst's report. Development of this approach could make a significant contribution to I&W DBMS analysis and evaluation.

Some related approaches are discussed in Section 15.6 of this report.

F.1.5 Decision Analysis

Decision analysis is based on a well-defined set of statistical methods for selecting among options, using assessments of the probabilities of contributing events, conditional probabilities (the probability of A, given that B occurs), and the expected gain or loss for each outcome. The approach is usually tied to Bayes' theorem, a formula for the relations among the probabilities.

Decision analysis systems have been widely advocated for use in major business, industrial, and military applications, although there have been difficulties in securing accurate assessments of the large number of probabilities required for practical use. Much of the current work in this area concentrates on the user interface, attempting to make it easier to secure probability assessments from persons who are knowledgeable in the subject matter but who are not expert statisticians.

Military intelligence analysis requires more effective methods for dealing with uncertainty in the information which it processes. Messages arriving at an intelligence center may be based on defector reports, prisoner interrogations, intercepted communications, and many other sources of information which may be intentionally deceptive. Electronic sensing devices, infrared sensors, and photographic equipment have finite limits of resolution, and the objects or events that they detect may be misidentified. Even when all available information is accurate, a potential enemy may change strategy or tactics at any time, in directions which are unpredictable even in principle. In short, intelligence information is always uncertain to some degree.

Although there has been some research in the measurement and reduction of uncertainty, there is no effective standardized method of measuring uncertainty or of combining uncertain information from heterogeneous sources. Decision analysis has been proposed as an effective technique for dealing with uncertainty, but it has not been widely used for this purpose. Instead:

o Intelligence is often reported with vaguely defined phrases like "it is believed that" or "it is somewhat probable that," which do not provide decision makers with enough information to support any clear conclusion.

o In some instances, two five-point scales representing accuracy and reliability are used; but research has shown that these scales are difficult to interpret reliably.

o Alternatively, numerical assessments are included, like "there is an 80 percent probability that" or "it is 60 percent likely that," where the numbers often represent intuitive estimates, rather than fully-supported assessments.

The use of sophisticated mathematical or statistical techniques can hardly be justified for information of this quality.

The contribution of decision analysis is in providing back-end processing for assessments of uncertainty. The Bayesian model is an effective method for combining probabilities from various sources to obtain overall probability assessments. In some instances (MYCIN is an example) rough numerical measures may be used, rather than well-defined probability assessments, because of difficulties in using Bayesian statistics with large data bases.

The major focus of current research, however, is on the front end, the user interface. The user may fill in blanks to create the probabilities and the tree structures required for full analysis. The user may return to earlier portions of the tree to correct faulty probability assessments, until a point is reached at which the decision structure is fully analyzed.

An area for research supported by the I&W DBMS analysis facility would be methods for attaching probability assessments to existing data. When information is retrieved under the DBMS, is there any effective way in which the user can determine the reliability or accuracy of the data? Does the DBMS have any way of dealing with uncertain or unreliable data?

The question is important, because the DBMSs typically accept data as accurate. In most applications, such as commercial or financial data processing systems, there is no provision for probabilistic information. A bank, for example, would never want to say, "There is an 80 percent probability that Jones still owes $1200 on his automobile loan." While there will inevitably be errors in data collection or entry, they do not include the type of uncertain information that is always present in intelligence data bases. For this reason, commercial DBMSs do not normally provide a capability for dealing with uncertainty.

Conceptually, it would not be difficult to include a field which would indicate the probability or the degree of uncertainty in I&W data. Since this information is rarely accurate to more than one digit at best (an exception would be sensor reports, where the accuracy of the sensor has been accurately determined), one byte would be more than sufficient to hold the probability assessment.

The more important question is the development of methods for making better use of uncertainty assessments. Often, they seem to function largely as hedges, to protect the analyst in case of error. If the analyst has said, "It is somewhat likely that Israel will attack Lebanon within the next week," then he or she cannot be accused of error, no matter what happens.

In the field of weather prediction, scoring rules are available to measure the accuracy of forecasts. When a weather forecaster announces, "There is a 40 percent chance of rain tomorrow," the scoring rule provides a statistically valid way of measuring the degree to which the forecast was right or wrong. The theoretical basis for these scoring rules has been extensively developed and is widely used for rating weather predictions.

There is no equally well developed theory for intelligence estimates, for a number of reasons:

- Intelligence analyses rarely have the kind of all-or-nothing confirmations that are found in forecasts of rain or snow. For this reason, it may be difficult to tell whether an intelligence analysis is confirmed or unconfirmed.

- Where weather forecasts are repeated hundreds or thousands of times, and thus can develop an adequate statistical basis for evaluation, intelligence estimates tend to be one-of-a-kind.

- While weather predictions have no effect whatever on the actual weather, the predictions of intelligence analysts can (and should) lead to actions which will affect the outcome. If an analyst predicts an enemy attack, the action of preparing for the attack may lead the enemy to change its plans, falsifying the prediction. This negative feedback effect may make it difficult to determine whether an intelligence analysis is right or wrong.

- In any case, little systematic work has been done in reviewing intelligence methods to determine which methods are most effective. There have, of course, been major studies of intelligence analysis in limited critical situations, such as the Pearl Harbor attack, but these studies do not consider the routine, day-to-day work of the I&W analyst. Without evaluation of past work, it is difficult to determine which types of analysis get the best results.

An extensive study of I&W analyses, with recommendations for the treatment of uncertainty, would be required for a full analysis of this problem. A less ambitious goal would be simply to determine whether there is some method, within the capabilities of current DBMSs, to indicate the uncertainty or probability of error in the data base.

It should be emphasized that this problem is not the same as the problem of maintaining data base integrity, which is essentially a problem for the DBMS (preventing accidental destruction of data, ensuring that all data remain accessible, avoiding inconsistencies in redundantly stored data, etc.) The problem we are considering here is that of uncertainty within the data items themselves, rather than possible errors in data management. The goal of the suggested experiments would be to determine whether uncertainty can be represented effectively within the data base.

F.1.6 Knowledge-Based Systems

Here, as in other paragraphs in this section, it is essential to concentrate on systems and methods which can actually be used by analysts working in the field, rather than on purely speculative systems of the distant future. Very recent results (since 1980) have indicated that AI has come out of the laboratory and into the real world. AI's practical importance has been signaled by the extensive commitment of corporations like Schlumberger and Texas Instruments to AI approaches in geological analysis for petroleum exploration, among many other examples.

A Knowledge-Based System (KBS) is essentially a compiler where the knowledge takes the form of rules or procedures, or a data base system where the knowledge is introduced as a list of facts. The KBS itself simply provides the software for effective access to knowledge. Knowledge is typically gathered from experts, who describe the heuristics or rules of thumb that they use for arriving at conclusions. Such a system is sometimes called an "expert system."

Some KBSs have been developed for specialized problem areas, such as medical diagnosis (MYCIN) and speech understanding (HEARSAY). These two systems have been generalized to permit them to be used with other types of knowledge. Other systems have been built from the start as the framework for general purpose KBSs. The Knowledge Representation Language (KRL) is an example of an approach to rapid development of a KBS, using expert knowledge as its base.

The line between KBSs and DBMSs is not sharply drawn; in fact, many of the DBMSs have capabilities which are comparable to those of KBSs. One direction for research would be to determine which of the well-known KBSs -- MYCIN, HEARSAY, etc. -- would be most useful in I&W analysis. Such a study would show the way in which advanced systems might be adapted to the operational environment.

Another approach would be to ask what facilities, comparable to advanced AI facilities, now exist in operational DBMSs, which are capable of handling the large data bases and rapid computation required by I&W analysts.

What capabilities might be included in a DBMS which would give it the flexibility and power of an AI system? Here are some examples (including several suggestions made earlier in this section):

o Access to multiple data bases in varying formats. A relatively small I&W center has more than 30 data bases available. At another center, a recent inventory included more than 100 data bases, some of which are very large. Each of these collections of data is stored in a different format, and is accessed in a different way. It should be possible to simplify access to data bases in different structures and formats.

- It should be possible for an analyst, using a single, very general query, to obtain information from any one of these data bases, with the system selecting the proper source. Moreover, it should be possible for one data base system to query another system when information is required, and combine the data extracted from more than one system to formulate an appropriate response to a single query.

- At no point should the user be forced to learn a new access language, or to devise a complex search strategy for obtaining information when it is needed.

- An essential task of I&W intelligence analysis is the detection and identification of anomalous events. From the time of Alexander the Great, military strategy has emphasized the need for surprise -- this is, for performing precisely those actions for which the adversary is unprepared. A self-organizing system will be one which rapidly reorganizes its structure to take account of anomalous events and to reduce the likelihood of surprise. In addition to responding to time-tested indicators of potentially significant events, it will locate areas of emerging interest -- meetings of high-level Soviet scientists, for example, or an increase in satellite surveillance -- even when these developments have not been pre-programmed by the analyst as potential indicators.

- Existing DBMSs are often difficult to use. Briefly-trained personnel on relatively short assignments cannot be expected to become computer experts. The use of many systems, however, requires that the analyst become virtually a computer programmer in order to obtain required information. An intelligent DBMS should make it possible for the analyst to enter a query in a simple, easily-learned language. When the analyst has become familiar with one set of techniques, it should not be necessary to learn a new accessing language with each new data base, but to continue to use the language and techniques which have already been learned.

o The system should anticipate the user's needs. Rather than a simple "yes" or "no" in response to a query, it should be able to provide the reasoning behind its response; it should be able to justify its answer.

o It should, moreover, be able to provide near-misses; if the user asks, "Are there Chinese tanks in Vietnam?," it should be able to reply, "No, but they're five miles from the border." Such facilities are now available only on AI systems with very small data bases. It should be possible to make them available for very large, complex data bases, like those encountered in intelligence applications.

o The intelligence community today is becoming increasingly aware of the need for quality control. Intelligence production is a difficult job, and the development of more effective techniques requires an assessment of the results of past intelligence products. Although the popular press has publicized some of the successes and failures of the intelligence community, what is required is a careful, day-by-day assessment of intelligence production, in order to provide the feedback required for improving intelligence methods. An intelligent DBMS will be self-assessing. It will provide techniques for reviewing past performance, comparing earlier predictions and estimates with the actual outcomes, and locating the areas in which failures have occurred. It will assist in locating the techniques or indicators which have been successful. It will learn from past experience. Such systems have long been a goal of AI research; similar techniques should now be applied to the assessment and improvement of operational systems for the I&W analyst.

- o Statistical methods for trend analysis, curve fitting, and other analytic techniques for the support of I&W have been developed. While many of these systems are extremely sophisticated, they have also proved to be difficult to use, particularly when the users have little background in the statistical methods involved. An intelligent DBMS should make it possible for a user to indicate a general problem or area of concern, with the system itself selecting the statistical methods, or other approaches, which are appropriate for the data. For example, the chi-square statistic should be used to determine whether the differences among certain measurements are significant, providing that the sample size is large enough, and that other constraints upon the data are met. For other applications, a nonparametric test may be used. But the I&W analyst should not have to be concerned with the subtle differences among the various statistical measures; the analyst should not even have to recognize that a statistical measure should be used. The system itself should be capable of making these choices, based on the characteristics of the data and the information requirements of the problem at hand.

This list represents a selection of the advanced facilities that I&W analysts could be given in a system which incorporates AI techniques. The I&W DBMS Analysis facility provides an environment in which such systems can be tested and demonstrated.

F.2 Data Base Machine Experiments

This section includes two subsections describing possible tests and demonstrations of data base machines:

- o The first set of experiments outlines Remote Terminal Emulation (RTE) tests for a Data Base Machine (DBM).

- o The second set uses the I&W DBMS facility for test and evaluation of a DBM for distributed I&W applications.

The purpose of this section is to provide concrete illustrations of the manner in which the I&W DBMS evaluation facility could be applied to further tests in support of I&W intelligence analysis.

F.2.1 RTE Tests of Data Base Machine

Data base machines have been introduced as potential replacements for software DBMSs. The purpose of this set of tests is to provide direct comparisons of the performance of an existing DBM through the use of benchmark tests developed for evaluating DBMSs.

Data base machines appear capable of offloading the data base management functions from a general-purpose CPU to a specialized processor, with a resulting increase in processing efficiency, as measured by such factors as response time, throughput, costs, and storage efficiency (i.e. disk space required for a given volume of data, disk I/O requirements for a sequence of accesses, etc.) These advantages may or may not be apparent in application to I&W intelligence data processing.

Experiments would include testing of a DBM, such as the Britton-Lee IDM-500, using the I&W DBMS scenarios, monitors, and data bases. The primary software development would be the adaptation of the scenarios and data bases for operation on the data base machine.

The I&W DBMS Analysis has provided measures of performance of three DBMSs against a set of scenarios. The outcome of this analysis is a set of performance parameters of the DBMSs, establishing a baseline against which performance of the DBM may be measured. To obtain these performance measures, the I&W DBMS Analysis has developed scenarios, data bases, drivers, and other software which can support testing of the DBM. This effort would use the results of the tests of software DBMSs, together with software developed for those tests, to determine whether the DBM provides a significant advantage for I&W data processing.

The first task would be the design of Remote Terminal Emulation (RTE) tests of the data base machine, developing software as needed to convert existing scenarios and data bases into formats required for inputs to the data base machine. The scenarios should be identical in structure to those previously used for testing software DBMSs. The goal of this task is to provide data which are comparable in form to the data obtained from the earlier tests of software DBMSs.

RTE tests would then be performed, providing performance statistics for the DBM. At least the following measures of performance could be obtained: response times, retrieval times, CPU times, overall (clock) times, number of disk accesses, disk head movements, and number of I/O requests processed.

A Live Test Demonstration (LTD) could also be prepared and executed. The purpose of this test would be to obtain qualitative factors which were not measured during the RTE, and to provide the basis for a demonstration of DBM capabilities. Such factors as ease of use, comprehensibility of the user language, time required for data preparation, and other human and qualitative factors could then be determined.

A weighting of evaluation factors, to ensure that the most significant features are given proper emphasis, would be prepared and applied to the results of the tests. These factors would be compared to identical factors obtained from the software DBMSs.

F.2.2 Distributed Data Base Machine Evaluation

Recent emphasis on distributed data bases and distributed computer processing in the I&W environment has raised questions of survivability of such systems. Traditionally, survivability has required the development of redundant facilities, at the cost of more than doubling hardware requirements. Data base machines may provide the basis for a lower-cost approach by eliminating the need for duplicating general purpose computers, through the use of specialized backup facilities for data base systems.

This project would develop designs for distributed I&W analysis facilities using offloading capabilities of data base machines for cost-effectiveness. Performance tests of a data base machine will be required to determine design parameters.

The project would include testing of a data base machine, as well as development of generalized system specifications for a distributed I&W data system. A detailed simulation employing data bases and procedures similar to those employed in the I&W environment would be required to determine design parameters.

It would begin with a study of survivability requirements for distributed data bases in the I&W environment. At least the following problems would be characterized and quantified: data volatility, data base size requirements, site locations, minimum transmission rates, security requirements, cryptographic transmissions, data integrity and consistency requirements. A detailed specification of I&W survivability requirements, as applied to distributed data base systems, would be the first product of this research.

An initial survivable system design could then be prepared, indicating those features and performance parameters that would be required to ensure a stated probability of survival of a I&W distributed data base under stated conditions of threat. The purpose of this initial design would be to identify performance requirements for hardware support for a survivable system.

A test plan would then be prepared. The I&W DBMS scenarios, data bases, monitors, and other software would be available for the tests. The purpose of the tests would be to obtain detailed, verified information concerning hardware performance for use in final system designs.

Following approval of the test plan, the tests would be carried out in accordance with the plan. Tests might include comparisons of performance of the DBM against performance of a general purpose computer system, to determine whether there were gains in performance and improved cost/performance ratios obtained through the use of the DBM. Problems identified during the initial study phase would be specifically addressed and potential solutions would be proposed and documented.

A report, including verified performance data, would then prepared indicating the applicability of a DBM to I&W distributed data bases, with emphasis on provisions for survivability. The report would include a recommendation based on the initial requirements study and on performance testing, concerning the contribution of specialized data base machines to the solution of the I&W survivability problem.

F.3 Knowledge Based Processor

The purpose of this section is to illustrate the type of research that might be supported by the I&W DBMS Analysis facility delivered to RADC at the conclusion of this project. This section will outline the goals and design constraints that are apparent at this time in the development of a general Knowledge Based Processor (KBP).

The KBP is intended as an interface between the user and a DBMS. This means that the commercial DBMS can be employed for storage and retrieval of data, and for many of the other functions that would ordinarily require detailed programming. On the other side is the interface with the user, who enters commands and queries. More importantly, the user suggests rules for the KBP to use in generating calls to the DBMS. The KBP accepts, stores, and processes these rules.

In this illustration, the user interface is not a "natural language" processor. This design decision is intended to serve two purposes: First, it relieves the developers of the task of designing and developing a natural language system, which is peripheral to the goals of DBMS analysis, extremely large, and quite risky (i.e., it might not work). Second, there is no convincing evidence that natural language is better, within the bounds of the tasks for which the KBP might be used, than a simple set of codes. The user can learn to use the codes just as quickly, and with as great a facility, as he or she can learn the constraints and conventions of any "natural language" processor that can reasonably be provided. For these two reasons, it will be assumed that the user language is a fairly simple set of commands, similar to SEQUEL.

Another design assumption is that a relational data base system will be used. This assumption is consistent with the recommendation of ORACLE in the I&W DBMS Analysis.

The KBP would use the scenarios developed for the I&W DBMS, together with the data bases that support them. In the end, the KBP is intended to be a generalized structure that would permit the storage and processing of rules for any type of data base, for any reasonable application; but the I&W software would provide a wide range of subjects which can be used for test and demonstration. It seems better to use existing material when the alternative would be to write new scenarios and data bases for demonstrations, which would be no more general than the I&W materials.

The following design components, then, would be used to support the KBP:

- Scenarios from the I&W DBMS Analysis.

- Data bases from I&W, at least for some initial tests. The general forms (i.e. the TIBS) would be more valuable than the data bases themselves, for the purpose of describing what the KBP would do.

- A DBMS operating under an existing system.

F.3.1 The Command Language

The format of the command language is taken to be SEQUEL-like. When the user types SEQUEL commands like:

```
SELECT  S#,STATUS
FROM    S
WHERE   CITY='PARIS'
```

The KBP recognizes them as legitimate SEQUEL commands and simply passes them through to the DBMS without processing. Responses from the DBMS to the user are also passed through, without processing. In other words, a user could use the DBMS in the normal way, without even knowing that the KBP functions were there.

Error handling would depend on the command. SEQUEL errors would be handled by the DBMS, and the KBP would have its own set of error handling procedures. Obviously, if a KBP command accidentally gets transmitted to the DBMS, or vice versa, it may be hard for the user to understand what's gone wrong.

The real work that the KBP does is to generate queries in SEQUEL, which it submits to the DBMS, and to intercept responses from the DBMS for further processing.

F.3.2 Implementation Language

The suggested language for implementation is C. Typically, artificial intelligence systems are implemented in some version of LISP, or in a LISP-based language like PLANNER. However, implementing the KBP in C would enhance its portability, at the possible expense of its ability to use existing AI programs.

F.3.3 Rules

What the KBP would do would be to process rules. Specifically, the user would enter a number of rules, representing "knowledge," over a period of time. These would be saved, probably by making use of the DBMS to allocate and store them like any other data file.

The rules would then be accessed by the KBP, which would retrieve them from the data base into a temporary file, and use them to perform specified operations. Typically, this means comparing data items against each of the rules, determining whether they match the elements of the left-hand side of the rule, and carrying out the designated operation if there is a match.

For example, suppose that a rule states: "If there are troop maneuvers when the commander is present on the battlefield, there is a high likelihood of attack." This rule generates a query to the data base, asking for the location of the commander whenever a troop maneuver is reported. If the commander's location is sufficiently close to the maneuver, a potential alarm is generated. The analyst then has the opportunity to review the potential alarm.

One problem will be to determine when to attempt the search. Clearly, once the KBP has generated a search for a specific item in the antecedent (the left-hand side) of the rule, there is no need to search for the same item again. It should not keep searching repeatedly through the same data. The search should be performed only when it is needed.

This suggests the need for demons that watch over new entries to the data base. A demon is a mask or matcher, which is applied to the stream of data entering the data base. When a data record is matched, a procedure is initiated, which performs some pre-specified job. Typically, the job would be· to carry out an additional search of the data base, to compare the results with results obtained by some other demon, to send a message to the terminal, or to do whatever else has been specified by the rule.

F.3.4 Data Structures

The data structure suggested for this implementation is not the usual network structure used by AI systems. Instead, a relational data base is used. This should mean a far less complex structure, and probably a less fragile one, than those usually used. The data bases that I&W analysts will be working with are more volatile than typical AI data bases.

Semantic nets are frequently used as a means of representing relationships and dependencies among data items in typical knowledge representation schemes. If a relational structure is to be used there should probably be some way of showing hierarchical and network structures, if the system is to be built on traditional AI procedures.

F.3.5 Demonstration

For an effective demonstration, there must be a set of rules for the system to exercise. Perhaps 100 rules will be needed for an initial demonstration. They might be obtained by tracing through the scenarios and identifying the reasoning processes that are used by the analysts in the scenarios. A place to start would be the Libyan scenario described in Section 7.0, which contains several examples of the analyst's reasoning.

The KBP described in this section illustrates the way in which the test facility developed for the I&W DBMS Analysis may be used to support test, analysis, evaluation, and advanced research in support of intelligence analysts.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.